NATURAL Version 2.3.2

Release Notes for Mainframes

Manual Order Number: NAT232–008IBB

This document applies to NATURAL Version 2.3.2 for mainframes, and to all subsequent releases.

Specifications contained herein are subject to change, and these changes will be reported in subsequent release notes or new manual editions.

TABLE OF CONTENTS

1. RELEASE NOTES 1

Documentation 1
- No New Manuals 1
- Correction to Version 2.3.1 Release Notes 1

System Commands 2
- LIST — New Functions for Selection List of Objects 2

Map Editor 2
- New User Exits in SYSEXT 2

Debugging Utility 3
- Availability 3
- Support of Steplibs 3
- Support of NATURAL OPTIMIZER COMPILER 3
- No Support of Old Versions 3
- Watchpoint Maintenance 3
- Commands 4
- File for Debug Environments 4
- State of a Debug Entry 4

Call Statistics Utility 5
- Availability 5
- Counter for Number of Calls 5

SYSMAIN Utility 5
- Location of Subprogram MAINUSER 5
- Deletion of File Security Profiles for DDMs 5

SYSRDC Utility 6
- Default Value of RDCSIZE Parameter Changed 6

SYSTP Utility 6
- New Functions for Thread Size and Roll File Size Calculations 6

SYSTRANS Utility 8
- Profile for Input Parameter Default Values 8
- SYSTRANS under NATURAL SECURITY 8

NATURAL for ADABAS SQL 9

NATURAL for DL/I 10
New Batch Utility for Selecting NDBs, NSBs and UDFs from a Dataset 10
NATURAL under CICS 12
New Parameter CHAP in Macro NCMPRM 12
NATURAL under CMS 12
Availability 12
Documentation 12
NATURAL under COM-PLETE 12
Print/Work File Access in Batch Environment 12
NATURAL under IMS/TM 13
Availability 13
Documentation 13
Prerequisites 13
Enhancements 14
Changes 15
Print and Work Files 22
Truncating and Padding Output Records under BS2000 22
Opening/Closing Print Files under COM-PLETE 22
User Access Method for Print and Work Files 22
Parameter Module 23
Database Specification for ENTIRE DB 23
Error Messages 23
Obsolete Error Messages 23
Support of LE/370 under VSE 23
Sysplex 24
Sysplex under CICS 24
Remote Procedure Call 24
Passing Floating-Point Parameters to/from Version 2.2 24
Size Limit for Data Elements 24
Dynamic Recataloging 25
Loading Datasets with INPL 25
Loading Old SOFTWARE AG Datasets 25
Initialization Errors with STACK=INPL 25

NATURAL SECURITY 26
Update INPL for Shared FSEC File 26
Copy User with Private Library 26
Expiration Dates of Version 2.2 Mailboxes 26
NATURAL ADVANCED FACILITIES 27
Use under IMS/TM 27
Spool File Conversion in Batch Mode 27
Defaults and Models 27
Reports 27
User Interface 28
NATURAL OPTIMIZER COMPILER 28
Dynamic Linking 28
NATURAL Connection 28
Communication Method SAG-NPA 28
NATURAL@Web 29
REVIEW NATURAL MONITOR 29

2. NATURAL UNDER IMS/TM 31

Environments 31
Interface Overview 32
IMS/TM Environments 33
Dialog-Oriented Environments 34
The Message-Oriented Environment 35
The Batch Message Processing Environment 39
Support of the NATURAL WRITE (n) Statement 40
Components 41
The Front-End Module 42
The NATURAL/IMS Interface Module NIIINTFM 43
The Physical Input Edit Routine 44
NIMDRIV Macro Parameters 45
NIMPARM Macro Parameters 46
NIMTRNTG Macro Parameters 56
NIMLPCB Macro Parameters 59
NIMMSGT Macro Parameters 60
NIMPIXT Macro Parameters 61

The Roll File and Roll Server 63
The Shared NATURAL Nucleus 64
The NATURAL Buffer Pool 64
The ADABAS Interface 65
The Preload List 65
Service Programs 66
Service Modules 79
Special Functions 94
Accounting 94
Monitoring 95
Broadcasting 97
The Multi-Session Feature 98
The Server Environment 100
User Exits 103
Recovery Handling 105
Abends (System and User Abends) 105
Non-Recoverable Errors 105
Recoverable Errors 105
Installing the NATURAL/IMS Interface 106
Prerequisites 106
The Installation Tape 107
Installing the NATURAL/IMS Interface 108
Installing the Multi-Session Feature (Optional) 123
Installation Verification 124
Error Codes 125

INDEX (FOR CHAPTER 2 ONLY) 149

1

RELEASE NOTES

These *Release Notes* inform you of the enhancements provided with Version 2.3.2 of NATURAL.

NATURAL 2.3.2 contains all ZAPs, early warnings and source changes applied to NATURAL 2.3.1 as error corrections.

This document summarizes the changes and corrections which result in a difference in handling between NATURAL Version 2.3.1 and Version 2.3.2.

If Version 2.3.2 is your first version of NATURAL 2.3 (that is, if you are migrating from NATURAL 2.2 to NATURAL 2.3.2), you should also read the *NATURAL Version 2.3.1 Release Notes for Mainframes*.

Documentation

No New Manuals

No new manuals are distributed with this release of NATURAL. A new edition of manuals is scheduled for the release of the next NATURAL version.

Correction to Version 2.3.1 Release Notes

The section **Other SOFTWARE AG Products with NATURAL 2.3** of the *NATURAL Version 2.3.1 Release Notes for Mainframes* lists the following product: “ADABAS SQL Server AIF Version 1.4.2”. This should read: “ADABAS SQL Server Version 1.4.2”.

System Commands

LIST — New Functions for Selection List of Objects

The LIST command provides two new functions, which can be invoked by entering the following new commands in the command line of the selection list of objects displayed by the LIST command:

- **Scan** — If you wish to list only those objects which contain a specific value, you enter the command **SC**. A window will then be displayed in which you specify the desired value, and also determine whether the scan is to be absolute or not. To deactivate this function again, you enter the command **SC OFF**.
  If SHORT (see below) is active, this scan function cannot be used.
- **"Short" List** — If you enter the command **SHORT**, only the "Name" and "S/C" columns will be displayed for the listed objects. To switch back to "normal" display (including all other columns), you enter the command **LONG**.

Map Editor

New User Exits in SYSEXT

For map profiles, the library SYSEXT provides two new user exits:

- **USR2016** can be used to read a map profile from the FNAT system file and store it on the FUSER system file.
- **USR2017** can be used to determine whether map profiles are to be read from the FNAT or FUSER system file. By default, they are read from FNAT.

For details, see the corresponding user exit texts in SYSEXT.

Debugging Utility

Availability

Version 2.3 of the NATURAL debugging utility, which was not available with the release of NATURAL Version 2.3.1, is now available with NATURAL Version 2.3.2.

With Version 2.3, the debugging utility provides the enhancements described below.

Note: The debugging utility, including all enhancements, is fully documented in the NATURAL Version 2.3 Utilities Manual for Mainframes.

Support of Steplibs

With Version 2.3, the debugging utility supports steplibs for breakpoints and watchpoints.

Support of NATURAL OPTIMIZER COMPILER

With Version 2.3, it is possible to apply the NATURAL debugging utility to programs compiled with the NATURAL OPTIMIZER COMPILER.

Whether it is possible or not to set breakpoints for lines compiled with the NATURAL OPTIMIZER COMPILER depends on the NODBG option of the OPTIONS statements.

No Support of Old Versions

Version 2.3 of the debugging utility can only be applied to Version 2.3 NATURAL objects, but not to objects cataloged with any previous version.

Version 2.3 of the Debugging utility can only maintain debug environments created with Version 2.3; debug environments created with any previous version will be ignored.

Watchpoint Maintenance

With Version 2.2, with a watchpoint operator other than MOD, program execution is interrupted when the specified condition is met for the first time. The next program interruption for the same watchpoint can only occur when the condition is met again *after* a variable change which caused the condition *not* to be met.

With Version 2.3, program execution is interrupted every time the condition is met and the variable content changes.

Commands

With Version 2.3, the debugging utility provides the following new commands:

- **OBJCHAIN** — At an interruption, this command displays the objects on the current level and all superior levels, as well as the current global data area (if applicable) and information on the interruption.
- **STEP SKIPSUBLEVEL** — When you enter this command upon a statement which invokes another object (for example, CALLNAT), processing is continued with the next statement line in the interrupted NATURAL object (instead of the first executed statement in the invoked object).
- **SYSVARS** — When you enter this command, the current values of various system variables are displayed.

The **SCREEN** command was inadvertently removed from the debugging utility description in the *NATURAL 2.3 Utilities Manual*. However, it is still available (with the same functionality as in Version 2.2: when you enter the SCREEN command upon interruption of a NATURAL object, the current screen of the interrupted object is displayed).

File for Debug Environments

With Version 2.2, debug environments are always stored in the system file FUSER.

With Version 2.3, an option in the user profile allows you to determine where debug environments are to be stored: in the system file FNAT, in the system file FUSER, or in the scratch-pad file.

State of a Debug Entry

With Version 2.2, the state of a debug entry can be either “A” (= active) or “P” (= pending).

With Version 2.3, “P” (= pending) has been renamed to “I” (= inactive).

Call Statistics Utility

Availability

Version 2.3 of the NATURAL Call Statistics utility (previously called Application Testpath Statistics utility), which was not available with the release of NATURAL Version 2.3.1, is available now with NATURAL Version 2.3.2.

Note: The Call Statistics utility, including all enhancements, is fully documented in the NATURAL Version 2.3 Utilities Manual for Mainframes.

Counter for Number of Calls

With Version 2.2, the format/length of the counter field for the number of calls was I2. With Version 2.3, it is I4.

SYSMAIN Utility

Location of Subprogram MAINUSER

The subprogram MAINUSER (which can be invoked to perform SYSMAIN functions directly from a user-written NATURAL application) must be located in a library whose name begins with “SYS” (but not in library SYSMAIN).

Deletion of File Security Profiles for DDMs

In a NATURAL SECURITY environment, the DDMs menu of SYSMAIN provides an option “Del. NSC–Def.”. If a DDM is deleted from the source environment or moved to a target environment with a different FSEC system file, you can use this option to determine whether or not the DDM’s file security profile is to be deleted from the source FSEC file.

SYSRDC Utility

Default Value of RDCSIZE Parameter Changed

The default value of the profile parameter RDCSIZE, which is used to activate/deactivate the SYSRDC utility and determines the size of its trace recording buffer, has been changed from 2 to 0. This means that, by default, SYSRDC is deactivated and has to be explicitly activated when desired (by setting the RDCSIZE parameter accordingly).

SYSTP Utility

New Functions for Thread Size and Roll File Size Calculations

The SYSTP utility now provides a new set of functions that allow you to determine an optimum thread size or roll file size for a NATURAL application under CICS, IMS/TM and UTM. (These functions are not available in a Sysplex environment.)

You should activate these functions only when needed, and deactivate them after you have determined your optimum thread size, because these functions occupy space in the NATURAL buffer pool. When you deactivate them, the space in the buffer pool becomes available again.

Proceed as follows:

1. Define an oversized thread in the range of 512 to 1024 KB for your NATURAL application. Take into account the number of SOFTWARE AG subproducts used.
2. Start your NATURAL application, either in production or in test mode.
3. Activate the NATURAL Thread Usage Statistics function: Invoke the SYSTP utility. On the SYSTP main menu, choose function “T” (NATURAL Thread Usage Statistics). On the menu that appears then, choose function “A” (Activate Statistics).
4. Use your NATURAL application under typical production conditions. The Thread Usage Statistics function runs in the background and logs the buffer sizes used.
5. Then invoke the SYSTP Thread Usage Statistics function again. On the menu that appears, choose function “S” (Show Statistics), “P” (Print Statistics) or “D” (Deactivate and Print Statistics). It is recommended that you use function “D” to free buffer pool space.

The NATURAL Thread Usage Statistics contain the following information:

Ext. Buffer	The sizes of these buffers are defined externally (in the NATURAL parameter module).
Defined Size	The buffer size as defined in the NATURAL parameter module.
Max. Allocated Size	The maximum buffer size allocated. Note that for the internal BB area, 14368 bytes are added to the ESIZE profile parameter value.
Max. Used Size	The maximum buffer size used.
Sum of External Buffer Sizes	The grand total of all buffer sizes defined in the NATURAL parameter module.
Sum of Internal Buffer Sizes	The grand total of all buffer sizes requested by NATURAL internally.
Max. Used Thread Length	The maximum thread length used by NATURAL. Define this length as your minimum (“optimum”) NATURAL thread length. Round it up to the next KB number that can be divided by 2.
Max. Compressed Thread Length	The maximum length of a compressed NATURAL thread that was written to the NATURAL roll file. Define this length as your minimum (“optimum”) NATURAL roll file length.

SYSTRANS Utility

Profile for Input Parameter Default Values

For the SYSTRANS utility, you can define a profile in which you can specify default values (both general and user-specific) for SYSTRANS input parameters. The corresponding fields in the SYSTRANS input screens/windows are then preset with these values. For the definition of the profile, the text object “PROFILE” is supplied in library SYSTRANS. To activate the profile, you have to save “PROFILE” under the name “TRANPROF” in library SYSTRANS.

For further details, see the source code of “PROFILE”.

SYSTRANS under NATURAL SECURITY

Under NATURAL SECURITY, the use of the SYSTRANS utility is controlled by utility profiles defined in NATURAL SECURITY (as described in the chapter **Controlling the Use of NATURAL Utilities** of the *NATURAL SECURITY 2.3 Manual for Mainframes*).

If no such profiles are defined for SYSTRANS, its use is controlled by the “old” utility protection mechanism (as described in the appendix of the *NATURAL SECURITY 2.3 Manual for Mainframes*). In this case, the description for SYSMAIN under “Maintenance Permission for Libraries” also applies to SYSTRANS.

NATURAL for ADABAS SQL

Together with Version 2.3.2 of NATURAL, Version 2.4.2 of NATURAL for ADABAS SQL is released.

It is described in a new edition of the *NATURAL for ADABAS SQL Manual*.

NATURAL for DL/I

New Batch Utility for Selecting NDBs, NSBs and UDFs from a Dataset

NATURAL for DL/I provides a new batch utilit — SELDLI — which allows you to *select* NDL objects (NDBs, NSBs, UDFs) from a dataset created by the ULDDLI utility.

The output of SELDLI can be used as input for the INPLDLI utility.

As INPLDLI does not allow for selection of objects from a dataset created by ULDDLI, you can use SELDLI to perform this function on desired objects prior to running INPLDLI. SELDLI can therefore be used for the backup/recovery or transfer of selected objects from test to production.

SELDLI also supports a scan (command SCN) feature that will list all of the objects on the input dataset without selecting any for output.

SELDLI can only be used in batch mode.

SELDLI requires the following input:

- the specification of the output dataset CMWKF01 from ULDDLI;
- up to 30 parameter lines containing the following:
  - object type (A3); the following types can be specified:
    NSB – select specified NSB,
    NDB – select specified NDB,
    NDU – select specified NDB and related UDF,
    UDF – select specified UDF,
    SCN – list input dataset CMWKF01,
    . – terminate SELDLI;
  - object name (A8); 1 occurrence.

Notes:
With NDB/NSB, a wildcard (*) can be specified at the end of the name to select a range of names.

*With UDFs, the object name must be in the form "nnn**nnn"; that is, a 3-digit database ID, followed by 2 asterisks, followed by a 3-digit file number.*

SELDLI provides the following output:

- A dataset containing selected objects to be used as input to INPLDLI. It is specified with DDNAME "CMWKF02".

When SELDLI is executed, the specified NDBs, NSBs and UDFs are copied from CMWKF01 to CMWKF02.

Example 1 — Select All NDBs:

```
LOGON SYSDDM
SELDLI
NDB,*
.
FIN
```

Example 2 — Select NSB “ORDPSB” and UDF for DBID 151, FNR 3:

```
LOGON SYSDDM
SELDLI
NSB,ORDPSB
UDF,151**003
.
FIN
```

Example 3 — Select NDB “CUSTDBD” and its Related UDFs:

```
LOGON SYSDDM
SELDLI
NSB,ORDPSB
NDU,CUSTDBD
.
FIN
```

Example 4 — List All Objects on the Input Dataset:

```
LOGON SYSDDM
SELDLI
SCN
FIN
```

NATURAL under CICS

New Parameter CHAP in Macro NCMPRM

The macro NCMPRM of the NATURAL/CICS parameter module NCIPARM contains a new parameter called CHAP (CHAnge task's dispatching Priority). This parameter defines how the NATURAL/CICS interface is to treat long-running tasks reaching the DBROLL and/or MAXROLL call limits.

- If NO is specified, the session is suspended (this is the default).
- If YES is specified, the task's dispatching priority is decremented by 1 every time it reaches the DBROLL and/or MAXROLL call limits. The original task dispatching priority is re-established at the next screen I/O.

NATURAL under CMS

Availability

Version 2.3 of the NATURAL interface for CMS, which was not available with the release of NATURAL Version 2.3.1, is available now with NATURAL Version 2.3.2.

Documentation

The NATURAL/CMS interface is described in the *NATURAL for CMS Manual*.

NATURAL under COM-PLETE

Print/Work File Access in Batch Environment

The module NATCMPL is used to access COM-PLETE print/work files in a batch environment; it has to be included in the linking of the NATURAL nucleus, together with the module COMPBTCH from the COM-PLETE distribution library.

NATCMPL was available with Version 2.2, but not with Version 2.3.1. With Version 2.3.2, it is available again.

NATURAL under IMS/TM

Availability

Version 2.3 of the NATURAL/IMS interface, which was not available with the release of NATURAL Version 2.3.1, is now available with NATURAL Version 2.3.2.

Documentation

As the NATURAL/IMS interface was not available with Version 2.3.1., it could not be documented in the *NATURAL Version 2.3 Installation and Operations Manual for Mainframes.*

Instead, a full description of the NATURAL/IMS interface is provided in Chapter 2 of these *Release Notes*.

With the next NATURAL version, the description of the NATURAL/IMS interface will be re-integrated into the *NATURAL Installation and Operations Manual for Mainframes*.

Prerequisites

NATURAL Version 2.3 requires IMS/TM Version 3.1 (or above), which requires MVS/ESA.

Enhancements

The NATURAL Version 2.3 interface to IMS/TM provides the following enhancements:

- The various NATURAL driver sources for the various IMS/TM environments have been combined in a single driver source.
- The NATURAL/IMS driver and the NATURAL/IMS interface module support RMODE ANY which allows NATURAL/IMS to be loaded above the line.
- The NATURAL/IMS driver and the NATURAL/IMS interface module are reentrant which allows them to be loaded into the LPA/ELPA.
- The error messages for non-recoverable error situations have been enhanced; the message texts are available in a modifiable text module.
- Error recovery for NATURAL has been enhanced (ESTAE instead of SPIE exit routine). This allows the release of pending ENQs in the case of an abend.
- The NATURAL thread may be obtained above the line except for in the BMP environment.
- The JES API is supported. This allows print files to be written directly to the MVS system spool.
- Support of stand-by server for all environments.
- The NATURAL roll server is supported which enables you to run NATURAL/IMS in a SYSPLEX environment.
- Support of IBM LE/370 Language Environment subprograms. The description provided in the *NATURAL Installation and Operations Manual for Mainframes* for MVS batch and TSO also applies to IMS/TM.

Changes

NIA and NII Consolidated

The NATURAL under IMS/DC Advanced Interface (NIA) no longer exists in its own right. The full functionality of NIA has been incorporated into the NATURAL/IMS interface (NII) Version 2.3.2.

NATURAL/IMS Interface Structure

All transaction codes for all environments are now specified in a single Transaction Code Table which is linked to the NATURAL/IMS interface module.

All Environment Tables are now specified in a single NATURAL/IMS parameter module which is linked to the NATURAL/IMS interface module. Each entry in the Transaction Code Table refers to one of the environment tables specified.

The name of the NATURAL/IMS interface module to be used is specified when generating the drivers.

The Environment Table / NII Parameter Module

The Environment Table of previous versions is now called the *NII Parameter Module*.

Parameters Removed

DIAGN	Obsolete as NATURAL error messages are used.
FLNMSG	The session start exit is now always invoked.
NATINTF	Obsolete due to NATURAL/IMS interface restructuring.
NATNUC	Replaced by a NATURAL profile parameter.
NWPCBS	Now part of the transaction code table.
SESPCB	Now part of the transaction code table.
SESTRN	Session switching is now PF-key driven.
SVC	Obsolete, as no swap pool available.
SWAPPL	Obsolete, as no swap pool available.
TRANSLT	Obsolete, as no swap pool available.
TRANTAB	Obsolete due to NATURAL/IMS interface restructuring.

The following parameters were only available with NIA and have now been removed:

FLNSPIE	Obsolete as ESTAE is used.
NIISVC	Functionality now provided by Authorized Services Manager.
SUBSYS	Functionality now provided by Authorized Services Manager.

Parameters Renamed

Old Name	New Name
ACTAHDR	ACTLHDR
LOG126	ERRLHDR
FLDBPCB	TERMDB
FLIPLFL	TERMIPL
NUMCOLS	COLPSCR
NUMLINE	LINPSCR
PBSIZE	HCBSIZE
RBSIZE	THSIZE, already done for NIA
SESACTV	MSACTV
SESDBD	MSDBD
SESMAX	MSSESMX

The following parameters were only available with NIA and have now been renamed:

Old Name	New Name
CMSIZE	CMBSIZE
FLSMF	ACTLOG

The Transaction Code Table

The layout of the transaction code table has been changed. You cannot use an existing Version 2.2 IGVCTAB translation code table with Version 2.3.

Parameters Removed

PCBS	Replaced by macro NIMLPCB.

Parameters Renamed

Old Name	New Name
PSB	PSBNAME

The Swap Pool Table

The swap pool table is obsolete as the swap pool is no longer available.

Monitor Pool Parameters

Monitor pool parameters are obsolete as the Authorized Services Manager SIP server is used for maintaining monitor data.

Broadcast Pool Parameters

Broadcast pool parameters are obsolete as the Authorized Services Manager SIP server is used for maintaining broadcast messages.

User Exits

All user exits have to be reassembled, because the layout of control blocks has been changed. Most fields of the IOCB extension and several fields of the scratch-pad area have been removed.

The following user exits are no longer available:

User Exit	Comment
NIIEMOD	Integrated in NIIXSSTA.
NIIIMSHC	Integrated in NIIXSSTA (see below).
NIIXT806	Obsolete as error situation no longer possible.

The following user exits have been renamed to adhere to naming conventions for user exits:

Old Name	New Name
SESSSTAX	NIIXSSTA.
NIISTART	NIIXSTAR. *Note: NIIXSTAR is invoked **after** the roll-in of the NATURAL thread, whereas NIISTART is invoked **before** it.*
NIIACTUE	NIIXACCT
NIIISRTX	NIIXISRT
NIIISRMX	NIIXISRM
NIISXIT	NIIXMSSP

Service Modules

The following service modules are no longer available:

Service Module	Comment
CMLASCB	Obsolete; internal use only.
CMMAPLST	Obsolete; internal use only.
CMSWPCOM	Obsolete as swap pool no longer available.

The following service modules are no longer documented and are only supported for compatibility reasons:

Service Module	Comment
CMPRNTR	Use *HARDCOPY instead.
CMTRNSET	Use CALLNAT CMPGMSET instead.

Swap Pool

The swap pool is no longer supported. The swap pool has been replaced by the *roll server.* For information on the roll server, see the section **The Roll File and Roll Server** on page 63.

Load Map

The load map is no longer required as 3GL programs are deleted at terminal I/O.

GETMAIN/FREEMAIN Table

The GETMAIN/FREEMAIN table is now part of base NATURAL.

The Bootstrap Module NIIBOOT

The bootstrap module for the message-oriented environment is no longer required, but it is still supported. The following parameters are obsolete:

Parameter	Comment
NATNAME	Replaced by the NATURAL profile parameter NUCNAME.
TRANNAM	The transaction code with which the bootstrap module was invoked is used.
PSBNAM	The PSB name of the bootstrap module is used.
NWPCBS	Now specified in the transaction code table.

The parameters can be specified but will be ignored.

Conversational Abnormal Termination Exit

The exit DFSCONE0 is obsolete as the NATURAL/IMS exit ESAE contains the required clean-up logic.

The BMP Control File

The BMP control file is optional. If it is used, the following parameters are obsolete and ignored during the session start:

Parameter	Comment
PSBNAME	The PSB name specified in the BMP JCL is used.
TRNCODE	The transaction code specified in the BMP JCL is used.
WRKPCBS	Now specified in the transaction code table.

The NATURAL profile parameters are evaluated if specified. If the CMPRMIN input is available, the NATURAL profile parameters in the BMP control file are appended. This means they will take priority over the CMPRMIN parameters.

The BMB Diagnose File

The BMB diagnose file is no longer supported. All non-recoverable error messages are written to the system log using the WTO macro.

The SPA Table (NIA)

The SPA table is obsolete and has been replaced by the Authorized Services Manager SIP server.

Print and Work Files

Truncating and Padding Output Records under BS2000

With Version 2.3.1, the keyword parameters TRUNC=ON/OFF (truncation of output records) and PAD=ON/OFF (padding of output records) in the NATURAL parameter module macros NTPRINT and NTWORK (or profile parameters PRINT and WORK respectively) were not supported under BS2000.

As of Version 2.3.2, these two keyword parameters also apply under BS2000.

Note: If errors NAT1510 and NAT1512 occur after you have transferred your existing NATURAL Version 2.2 applications to Version 2.3.2, you should set TRUNC=ON and PAD=ON.

Opening/Closing Print Files under COM-PLETE

For the print file access method AM=COMP, the default values of the keyword parameters OPEN and CLOSE in the NATURAL parameter module macro NTPRINT (or profile parameter PRINT) have been changed to be compatible with the behaviour in NATURAL Version 2.2. The default values are:

- Version 2.3.1: **AM=COMP,OPEN=OBJ,CLOSE=FIN**
- Version 2.3.2: **AM=COMP,OPEN=ACC,CLOSE=OBJ**

User Access Method for Print and Work Files

With NATURAL 2.3.2, a new module NATAMUSR and a new macro NAMAMUSR are delivered. The NATAMUSR module provides an exit interface (entry point NATAM9EX) for software vendors to handle NATURAL print and work files. The NATAMUSR module (with the access method exit) may be installed in one of the following ways:

- linked to the NATURAL nucleus,
- linked to the NATURAL driver (when driver and front-end are split),
- linked to an alternative NATURAL parameter module (as loaded via profile parameter PARM),
- linked as a separate module; in this case, the following NATURAL profile parameters are required: RCA=(NATAM09), RCALIAS=(NATAM09,*xxx*) where *xxx* is the name of the separate module in the load library.

For a NATURAL print/work file to be handled by the user access method, AM=USER has to be set for the file in the NTPRINT or NTWORK macro (or PRINT or WORK profile parameter respectively) of the NATURAL parameter module. For hardcopy printers, the profile parameter HCAM=USER has to be set.

Parameter Module

Database Specification for ENTIRE DB

With NATURAL Version 2.3, the way in which a database to be handled by ENTIRE DB is specified in the NTDB macro of the NATURAL parameter module is different from Version 2.2:

- Version 2.2: **NTDB ENTIRE,***database-ID*
- Version 2.3: **NTDB ADAV***n*,*database-ID*,**ENTIRE**

Error Messages

Obsolete Error Messages

The following error messages have become obsolete; they no longer exist with Version 2.3: NAT9000, NAT9100, NAT9101 and NAT9200.

Support of LE/370 under VSE

The support of IBM LE/370 Language Environment subprograms, which has been made under MVS with NATURAL Version 2.3.1, is now also available under VSE.

NATURAL prepares the CALL statement which invokes these subprograms to make them executable under LE/370. LE/370 subprograms can be static or dynamic subprograms of NATURAL. For details, see the *NATURAL Installation and Operations Manual for Mainframes*.

Sysplex

Sysplex under CICS

When the NATURAL/CICS interface is set up with NCIPARM generation parameter COMARET=NO, or when the NATURAL/CICS interface has been invoked by a front-end program via EXEC CICS LINK (see also *NATURAL Installation and Operations Manual for Mainframes*, page 546), the NATURAL/CICS interface's session restart information is written into CICS temporary storage (and not into the CICS COMMAREA).

In a Sysplex environment, this requires that the CICS temporary storage key prefix used for the NATURAL session restart information (as specified as second value of the NCIPRM parameter TSKEY) be defined in a CICS temporary storage table (TST) as REMOTE/SHARED in all participating CICS application regions.

Remote Procedure Call

Passing Floating-Point Parameters to/from Version 2.2

If floating-point parameters are passed in a remote procedure call to/from a partner (client or server) whose NATURAL version is 2.2, the profile/session parameter DC in this Version 2.2 NATURAL must be set to "." (period); otherwise, conversion errors will occur.

Size Limit for Data Elements

As of NATURAL Version 2.3.1, the maximum size of a global or local data area can be 16 MB (as stated in the *NATURAL Version 2.3.1 Release Notes for Mainframes*).

The size of a single data element (array or indexed group) within a global/local data area, however, must not exceed 32 KB. As of Version 2.3.2, error NAT0476 will be issued at compilation if this size is exceeded.

Dynamic Recataloging

The profile parameter RECAT only applies to Version 2.3 objects. An error message is issued if RECAT=ON detects an inconsistency concerning a Version 2.2 program and/or global data area.

Loading Datasets with INPL

Loading Old SOFTWARE AG Datasets

As stated in the *NATURAL Version 2.3.1 Release Notes for Mainframes*, the loading of datasets into the system files with INPL is restricted to datasets that are identified as official SOFTWARE AG INPL system datasets.

However, some old SOFTWARE AG INPL datasets may not yet have the appropriate internal identification, and their loading will be rejected with an INPL error message. To be able to load such datasets, the following temporary solution is provided: Link the INPL module to the shared NATURAL nucleus, and apply special-purpose ZAP NA31113.

Initialization Errors with STACK=INPL

If a NATURAL session is started with the profile parameter STACK=INPL, any session initialization error would cause the session to be terminated immediately — regardless of the setting of the profile parameter ITERM.

NATURAL SECURITY

Update INPL for Shared FSEC File

As a prerequisite for an FSEC system file to be shared by NATURAL SECURITY Versions 2.2 and 2.3, the *NATURAL SECURITY Version 2.3 Manual for Mainframes* states update INPL number “NE2839”, whereas the *NATURAL Version 2.3.1 Release Notes for Mainframes* state update INPL number “NE2841”. The correct update INPL to be applied to the Version 2.2 NATURAL environment is “NE2841”.

Copy User with Private Library

As of NATURAL SECURITYVersion 2.3.2, when an existing user security profile (of user type A or P) is copied, the private library (if defined) will also be copied into the new security profile.

Expiration Dates of Version 2.2 Mailboxes

Mailboxes created with NATURAL SECURITY Version 2.2 have expiration dates with 2-digit year components *nn*. If you use these mailboxes under Version 2.3, these expiration dates are assumed to be 19*nn*.

If this is not desired, that is, if an expiration date is to be 20*nn*, you have to convert the expiration date to one with a 4-digit year component.

For the purpose of converting mailbox expiration dates, library SYSSEC provides the program program CHCKNSC. When you execute this program, a menu is displayed, on which you select function “C”. For details on this function, refer to its online help (invoke the function, then press PF1).

NATURAL ADVANCED FACILITIES

Use under IMS/TM

For the use of NATURAL ADVANCED FACILITIES under IMS/TM, the definition of printers was expanded to include:

- the indication whether the printer is a SCS printer (Y/N);
- the maximum size of the buffer used to send data to the printer (this size must be in the range from 256 to 4048 bytes).

Spool File Conversion in Batch Mode

If the conversion of the spool file from NATURAL ADVANCED FACILITIES Version 2.2 to Version 2.3 is performed in batch mode, the printers cannot be automatically assigned to a specific operating/TP system. The printers are then marked with "*BATCH", and must be assigned online after the spool file conversion.

For this purpose, function 32.3 (Mass Update for Printers) has been enhanced: It now allows you to assign all printers marked with "*BATCH" to a valid system (CICS, IMS or BS2000). In the case of a spool file used by several different systems, you have to use function 31.4 to assign the printers individually.

Defaults and Models

The display of function 30.5.2 was changed so that object types for which a model already exists are displayed intensified.

For the definition of a default logical printer, the following applies:

- If a definition exists, the function text is set to "Modify".
- If no definition exists, the function text is set to "Define".

Reports

To avoid the problem of "mixed reports", the connection between index and report was expanded to also include a timestamp. If the timestamp values differ, a corresponding error message will be issued.

It is therefore strongly recommended that the spool file be formatted when you change from Version 2.3.1 to Version 2.3.2 of NATURAL ADVANCED FACILITIES.

User Interface

The layout of the NATURAL ADVANCED FACILITIES user interface has been slightly revised.

Note: These user interface revisions will be reflected in the next edition of the NATURAL ADVANCED FACILITIES Manual.

NATURAL OPTIMIZER COMPILER

Dynamic Linking

The NATURAL OPTIMIZER COMPILER can be linked statically to the NATURAL nucleus (as described in the *NATURAL OPTIMIZER COMPILER Version 2.3 Manual*). As of Version 2.3.2, it can also be linked dynamically during the initialization of the NATURAL session. If it is to be linked dynamically, the profile parameter RCA=(NOCNUC) has to be specified.

NATURAL Connection

Communication Method SAG-NPA

The communication method SAG-NPA is no longer supported by NATURAL Version 2.3.

NATURAL@Web

Together with NATURAL Version 2.3.2, a new SOFTWARE AG product is available: Natural@Web.

With NATURAL 2.3.2 and Natural@Web, you can create web pages via a NATURAL subprogram.

This enables you to:

- return dynamic web pages generated by NATURAL subprograms,
- access the HTTP interface of your HTTP server (cookies),
- return different kinds of documents containing alphanumeric data,
- use predefined programs for HTML generation.

NATURAL provides three new libraries containing the NATURAL web interface, which is called by the EntireX web adapter.

The program NAT–STLB is supplied to allow you to set up the required steplib hierarchy.

For details on Natural@Web, please refer to the Natural@Web documentation.

REVIEW NATURAL MONITOR

A new version of the REVIEW NATURAL MONITOR, Version 3.5.1, is released at the same time as NATURAL Version 2.3.2. This new version supports NATURAL Versions 2.2 and 2.3.

For details, please refer to the REVIEW NATURAL MONITOR documentation.

2

NATURAL UNDER IMS/TM

This chapter describes how to run NATURAL under IMS/TM and contains the following sections:

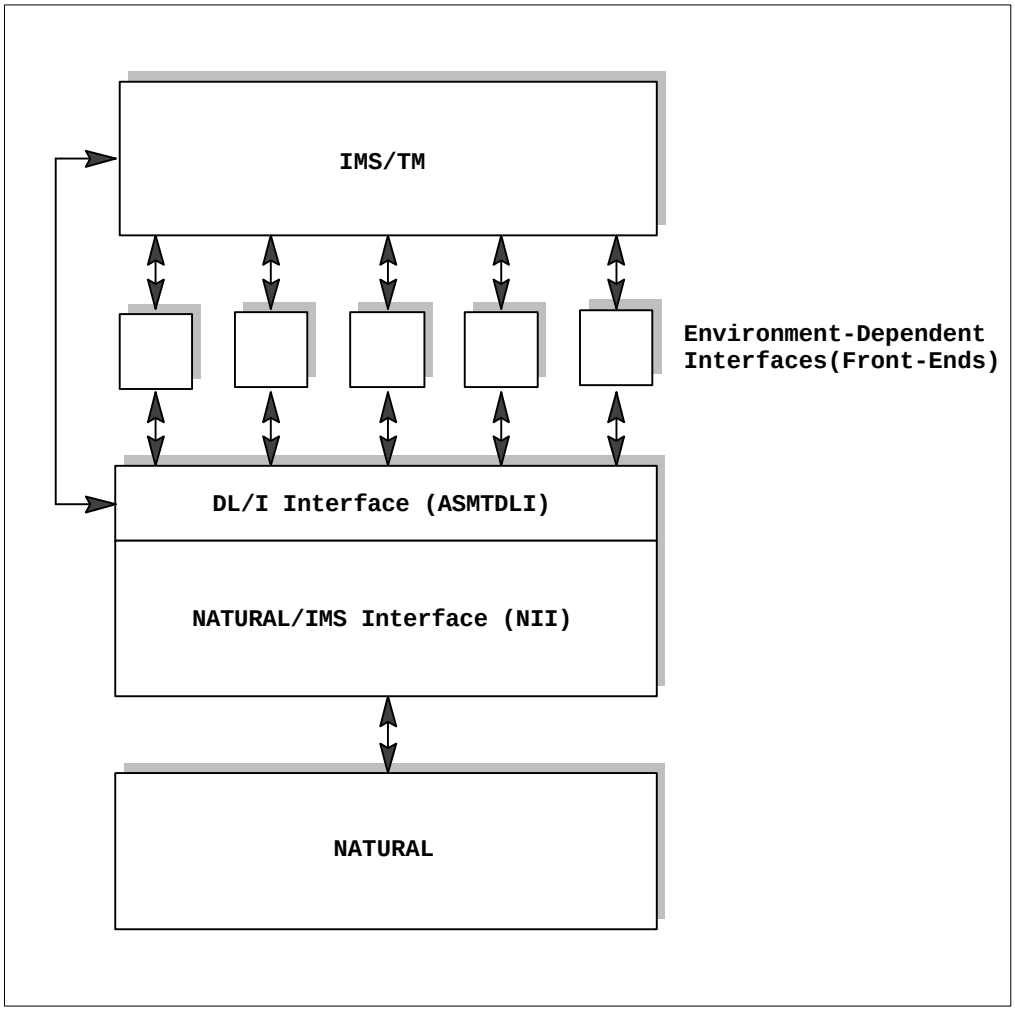
- Environments (page 31)
- Components (page 41)
- Special Functions (page 94)
- User Exits (page 103)
- Recovery Handling (page 105)
- Installing the NATURAL/IMS Interface (page 106)
- Error Codes (page 125)

Environments

This section describes how NATURAL runs under various IMS/TM environments and contains the following topics:

- Interface Overview (page 32)
- IMS/TM Environments (page 33)
- Dialog-Oriented Environments (page 34)
- The Message-Oriented Environment (page 35)
- The Batch Message Processing Environment (page 39)
- Support of the NATURAL WRITE (*n*) Statement (page 40)

Interface Overview



IMS/TM Environments

IMS/TM provides three different types of environments:

- message processing regions,
- batch message processing regions,
- IMS batch jobs.

To be able to use NATURAL in each of these environments, different environment-specific interfaces are provided for the NATURAL/IMS interface. The task of such an interface is to receive input (usually a terminal input message) from the environment, to pass the input to NATURAL for processing and to direct the resulting output back to the correct destination (usually a terminal output message). This way, it is possible to use the functionality of NATURAL in all available IMS environments.

In addition to different available environments, within each environment, there are different ways of operating.

NATURAL in a Message Processing Region (MPR)

In a message processing region, NATURAL online transactions can be one of the following:

- dialog-oriented,
- message-oriented.

Dialog-Oriented NATURAL

A dialog-oriented NATURAL session establishes an ongoing interaction with an IMS screen. Input and output messages to and from NATURAL are logically related and, across dialog steps, NATURAL saves information so as to be able to correctly process the next input message. In a dialog-oriented way, NATURAL can be run as either a conversational or a non-conversational transaction.

In a dialog-oriented environment, NATURAL can be executed in multiple-message processing regions, as Wait-for-Input (WFI) transaction and with the parallel-scheduling option.

To run NATURAL in dialog-oriented environments, you either have to use the roll server or roll files (see **The Roll File and Roll Server**, page 63).

If the NATURAL/IMS interface detects an error situation, a record containing information about this error situation is written to the IMS log file (see **Recovery Handling**, page 105). Thus, all terminals on which NATURAL is to be executed and all NATURAL transaction codes have to be authorized to issue the /LOG command using the automated operator interface (AOI).

Message-Oriented NATURAL

A message-oriented NATURAL session processes non-3270-formatted messages from the IMS message queue. The input messages are considered to be unrelated to each other and are not part of a dialog. In a message-oriented way, NATURAL must be run as a non-conversational transaction.

NATURAL in a Batch Message Processing (BMP) Region

In a batch message processing region, NATURAL can have access to the IMS message queue by using an input transaction code. With batch-oriented BMP regions, NATURAL supports symbolic checkpoint and extended restart. The input messages are non-3270-formatted messages.

NATURAL in IMS Batch Mode

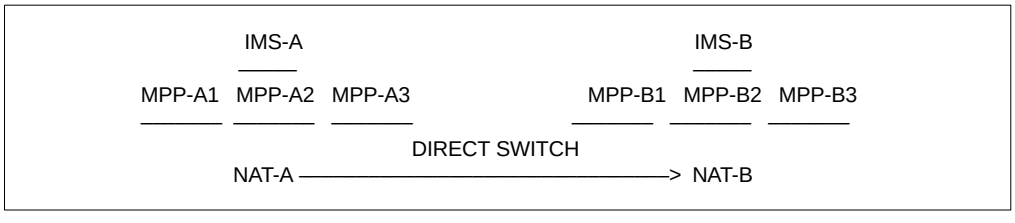
The BMP NATURAL can also be executed as an IMS batch job.

Dialog-Oriented Environments

This section discusses special points valid for the dialog-oriented conversational environment only. The dialog-oriented conversational environment requires a scratch pad area (SPA) of at least 173 bytes. It simulates the SPA by using the SIP server component of the Authorized Services Manager.

Special Considerations for an MSC Environment

Assuming the following environment, the NATURAL/IMS interface prepares the message X'000500006D' for NAT-B, which means that the terminal user has pressed CLEAR.



Two entries must be created in the transaction code table: the first entry is for NAT-A, the second for NAT-B.

These two entries must specify different offsets for the NATURAL reserved area (NRA) and must ensure that these areas do not overlap.

NAT-B detects that a NATURAL session is to be started in IMS-B in the usual way and therefore gives control to its session-start exit routine. The session-start exit routine checks the input message for the string X'000500006D' and sets to "0" the length of the input message as seen by NATURAL.

If no additional logic is provided in either the exit NIIXSTAR or the exit NIIXSSTA, NATURAL starts a new user session in IMS-B.

It is assumed that IMS-A and IMS-B have different dedicated roll files allocated for NATURAL.

Both (or more) NATURAL sessions can communicate with each other by transferring data in the SPA when performing direct program-to-program switching.

Note: For the time being, when two or more NATURAL sessions exist in such an environment, only the "active" session is terminated correctly.

The Message-Oriented Environment

This section describes the message-oriented interface for use with NATURAL for IMS/TM.

Introduction

This interface is designed to process nett-data input messages, which means that the messages do *not* represent a 3270 data stream. The message-oriented interface is driven by a user-written NATURAL program which instructs the interface to access the IMS message queue for the purpose of retrieving input messages.

The message-oriented interface has been created to support non-conversational, non-terminal driven transactions which must be executed as non-conversational MPP transactions.

Operation

The message-oriented interface incorporates functions from both the MPP and the BMP interfaces. The BMP interface is used as a basis, since much of the processing required emulates BMP-type transactions.

Since the message-oriented interface is not terminal-oriented, no messages or screen images are automatically generated to be sent to a terminal. The NATURAL nucleus is informed that it is running in a batch environment; therefore output is interpreted to be printer output and input is expected from a CMSYNIN file. All output which is normally written to CMPRINT is sent to the IMS/TM destination (normally an LTERM) specified with the NATURAL profile parameter SENDER. If NATURAL attempts to retrieve input data and no input data has been supplied by the application through the STACK command, return code 4 indicates that no input exists and NATURAL is terminated.

You can set SENDER to a new value at runtime by using the service module CMSNFPRT.

Except for checkpoint processing, NATURAL for DL/I and NATURAL for DB2 process as if they were in BMP mode. This is necessary, since one physical scheduling can (and usually will) process several unrelated input messages. Under the conversational MPP interface, all transactions processed during one NATURAL session and all DL/I requests within this NATURAL session are considered to be related, requiring maintenance of database positioning and PCB usage. With the non-conversational interface, this NATURAL for DL/I logic is not applicable.

Since transactions which are processed during one scheduling (and one NATURAL session) are not related to each other, the retention of NATURAL session information in the roll file is not required. Thus, no roll dataset needs to be allocated for this interface. A roll slot area is still allocated via GETMAIN and used to store all NATURAL control blocks and work areas.

Since processing is performed on a message-by-message basis, there is no need for any relocation logic.

With the message-oriented interface, retrieval of all messages from the message queue is initiated by a front-end NATURAL program. This program must be user-written to meet your specific processing needs. However, it requires a specific structure, as shown in the following:

```
PROGRAM INITIALIZATION
REPEAT
  CALL 'CMGETMSG' MESSAGE-AREA MESSAGE-LENGTH
  IF MSG-LL = 0                          /*  QC on GU to message queue
    TERMINATE
  FETCH RETURN PGMA MESSAGE-AREA
  REPEAT
    CALL 'CMGETSEG' MESSAGE-AREA MESSAGE-LENGTH
    IF MSG-LL = 0                        /*  QD on GN to message queue
      ESCAPE
    FETCH RETURN PGMB MESSAGE-AREA
  END-REPEAT
END-REPEAT
END
```

The service module CMGETMSG reads the first message segment. The service module CMGETSEG reads all further message segments.

Since NATURAL cannot read input from CMSYNIN, it is required to use the NATURAL stack for input. This is done by using the NATURAL profile parameter STACK.

It is your responsibility to ensure that the IMS message queue is accessed by your application prior to the termination of NATURAL. If not, the NATURAL transaction abnormally ends with IMS abend code 462, indicating that a GU to the message queue has not been performed.

To obtain these NATURAL messages even in the case of an abnormal termination, you are recommended to define the first alternate PCB as an EPRESS PCB.

The message-oriented interface can either be called directly by the IMS/TM program controller or via a bootstrap module.

If it is called directly by IMS/TM, it is recommended to use a NATURAL profile parameter which contains the required STACK parameter. Specify PROFILE=PROGRAM in your NATURAL parameter module and create a profile with a name equal to the transaction code with which the interface is invoked. This way, you have the flexibility to use a different profile with a different STACK for each transaction code used.

If it is called via a bootstrap module, the bootstrap module provides a string of dynamic profile parameters, one of which is the STACK profile parameter.

The Bootstrap Module NIIBOOT

The bootstrap module is called by the program controller DFSPCC20. It loads and calls the message-oriented front-end module passing the necessary parameters. The bootstrap module NIIBOOT is delivered as source module and has to be assembled and link-edited during the installation process. The source code is contained in the macro NIMBOOT. NIMBOOT includes the following parameters:

DRIVERN

Possible values:	Any valid MVS module name
Default value:	None

Specifies the name of the front-end module.

ENVTNAM

Possible values:	Any valid MVS module name
Default value:	None

Specifies the name of the environment table. This parameter is optional. If it is not specified, the environment table is determined by the entry in the transaction code table which corresponds to the transaction code used.

DYNPARM

Possible values:	Any character string of up to 80 characters.
Default value:	None

DYNPARM is used to define a valid string of up to 80 characters of NATURAL dynamic parameters.

The Batch Message Processing Environment

A standard batch NATURAL is executed in a Batch Message Processing (BMP) region. In comparison with the standard batch NATURAL, the optional input dataset CONTROL may be used.

The BMP CONTROL File

The optional BMP CONTROL file contains a maximum of two input cards. The first input card must contain the following keyword:

Keyword	Meaning
ENV-TAB=	The name of the environment table to be used.

Example:

```
ENV-TAB=ENVBMP0
```

The second input card of the CONTROL file contains the dynamic NATURAL parameters. If the CMPRMIN dataset is also used to pass dynamic NATURAL parameters, the input of CONTROL is appended to the input of CMPRMIN. This means the parameters specified in CONTROL overwrite the parameters specified in CMPRMIN. If the CONTROL file is not used, the name of the environment table is determined by the entry in the transaction code table which corresponds to the transaction code used (transaction-oriented BMP) or to the PSB name used (batch-oriented BMP).

Support of the NATURAL WRITE (*n*) Statement

With the WRITE (*n*) statement, up to 31 different reports on different printers can be produced within the same NATURAL program. The reports are sent to the IMS terminals specified either in the NATURAL parameter module or by using the NATURAL DEFINE PRINTER (*n*) statement. You have to specify AM=IMS in the NTPRINT macro which controls the report.

To be able to use this statement, define as many additional alternate TP-PCBs in your PSB as the number of parallel reports you want to create within the same NATURAL program, and specify the number of additional alternate TP-PCBs in your transaction code table by using the parameter WRKPCBs.

Note: Be aware that the first alternate TP-PCB is used by the NATURAL/IMS interface.

When using the WRITE (*n*) statement in a dialog-oriented environment, the following restriction applies:

The generation of a report cannot span one or more screen I/Os. If you want to use the same printer after a screen I/O, you have to close it explicitly before the screen I/O using the CLOSE PRINTER(*n*) statement.

To create reports, the following parameters of the NTPRINT macro are relevant:

Parameter	Meaning
AM	Must be set to “IMS”.
DEST	Specifies the IMS/TM destination.
BLKSIZE	Specifies the size of the buffer which is sent to the destination. Report lines are buffered.
DRIVER	Specifies the driver to be used to create the report. For a list of possible values, see the PRTDRIV parameter of the NATURAL/IMS parameter module. The driver determines where you want to have the form feed (at the start of the report, the end, both the start and the end, or no form feed), where you want to start your page (on line 1 or on line 2 for NII 2.2 compatibility) and where you want to print your report (SCS or non-SCS printer). In addition, you can specify that you want to use the JES API.
NAME **FORMS** **DISP** **COPIES** **CLASS** **PRTY**	Are only evaluated if you use the JES API.

Components

This section discusses the components of the NATURAL/IMS interface. The following topics are covered:

- The Front-End Module (page 42)
- The NATURAL/IMS Interface Module NIIINTFM (page 43)
- The Physical Input Edit Routine (page 44)
- NIMDRIV Macro Parameters (page 45)
- NIMPARM Macro Parameters (page 46)
- NIMTRNTG Macro Parameters (page 56)
- NIMLPCB Macro Parameters (page 59)
- NIMMSGT Macro Parameters (page 60)
- NIMPIXT Macro Parameters (page 61)
- The Roll File and Roll Server (page 63)
- The Shared NATURAL Nucleus (page 64)
- The NATURAL Buffer Pool (page 64)
- The ADABAS Interface (page 65)
- The Preload List (page 65)
- Service Programs (page 66)
- Service Modules (page 79)

The Front-End Module

The front-end module receives control from the IMS/TM program controller DFSPPC20, except in the server environment where it is called by the call interface NIIBOOTS (see page 100).

The front-end module must be created during the installation process and consists of the following:

- environment-dependent interfaces (drivers),
- the NATURAL parameter module,
- the NATURAL work file access method for the BMP environment,
- modules from other NATURAL products.

The Environment-Dependent Interfaces (Drivers)

You must generate an environment-dependent interface for each IMS environment supported by NATURAL/IMS using the NIMDRIV macro.

For a detailed description of the macro NIMDRIV, see page 45.

The NATURAL Parameter Module NATPARM

For information on the NATURAL parameter module, see the *NATURAL Installation and Operations Manual for Mainframes*.

The Work File Handling Module NATWKFO

The NATWKFO module is delivered as part of the base NATURAL. It is used for work file and print file handling in the BMP environment only.

Modules from Other NATURAL Products

Some NATURAL products, such as NATURAL for DB2 and NATURAL for DL/I, require that their modules be linked to the NATURAL/IMS front-end module. For further information, see the appropriate product documentation.

The NATURAL/IMS Interface Module NIIINTFM

The NATURAL/IMS interface module has to be created during the installation process and is common to all environments.

The interface module consists of the following components:

- NATURAL/IMS nucleus,
- NATURAL/IMS parameter module NIIPARM,
- transaction code table NIITRTAB,
- message text module NIIMSGT,
- DL/I language interface ASMTDLI.

The NATURAL/IMS Nucleus

The NATURAL/IMS nucleus is delivered as a load module and contains all the runtime routines required by the NATURAL/IMS interface.

The NATURAL/IMS Parameter Module NIIPARM

The NATURAL/IMS parameter module NIIPARM contains a number (1 – *n)* of parameter tables each defined by the macro NIMPARM and identified by the parameter ENTRYNM. Each parameter table within NIIPARM defines the NATURAL/IMS-specific parameters for a particular environment. Thus it is possible to set the parameters for all NATURAL/IMS environments in one parameter module. The environment which is currently used is set in the transaction code table NIITRTAB.

For a detailed description of the macro NIMPARM, see page 46.

The Transaction Code Table NIITRTAB

The transaction code table NIITRTAB table is of variable length and each entry is generated by the macro NIMTRNTG.

Each entry in the transaction code table refers to an entry in the NATURAL/IMS parameter module and can be followed by one or more occurrences of the macro NIMLPCB.

For a detailed description of the macro NIMLPCB, see page 59.

The appropriate entry within this table is detected by the current transaction code. If, in a non-message-driven BMP, no transaction code is defined, the current PSB name is taken instead.

For a detailed description of the macro NIMTRNTG, see page 56.

The Message Text Module NIIMSGT

The message text module NIIMSGT is part of the NATURAL/IMS interface module and is supplied both as a load and a source module. For each possible NATURAL/IMS runtime error, it contains the corresponding message text. Each entry is generated by the macro NIMMSGT.

For a detailed description of the macro NIMMSGT, see page 60.

The DL/I Language Interface ASMTDLI

The DL/I language interface ASMTDLI is part of IMS/TM.

The Physical Input Edit Routine

The physical input edit routine is required only in a dialog-oriented, non-conversational environment. It is used to insert the transaction code preceding the message sent to the terminal. This is required as NATURAL runs in MFS bypass mode and the message sent to the terminal does not contain a transaction code.

The physical input edit routine is generated by using the NIMPIXT macro. For futher information on the NIMPIXT macro, see page 61.

Once the physical input edit routine is generated, its name must be specified in the TYPE or LINEGRP macros of your IMS/TM system definition. For all terminals on which the non-conversational environment is supposed to run, you must enable physical editing by using the EDIT parameter in the TERMINAL macro.

NIMDRIV Macro Parameters

The macro NIMDRIV generates environment-dependent interfaces (drivers).

The parameters which can be specified with the macro NIMDRIV are described below:

TYPE

Possible values:	
TYPE=CONV	Dialog-oriented conversational environment generated.
TYPE=NONC	Dialog-oriented non-conversational environment generated.
TYPE=NTRD	Message-oriented (not terminal-driven) environment generated.
TYPE=BMP	Batch-oriented and transaction-oriented BMP environment generated. *Note: You can also use this interface in the DLIBATCH environment.*
TYPE=SRVD	Server environment generated.
Default value:	None

LE370

Possible values:	
YES	The LE/370 environment is initialized and remains so until the NATURAL/IMS front-end returns to the IMS program controller.
NO	The LE/370 environment is not initialized. An LE/370 environment is initialized and terminated for each LE/370 program call.
Default value:	NO

Specifies whether NATURAL/IMS initializes the LE/370 environment.

NIINAME

Possible values:	Any valid module name up to 8 characters.
Default value:	NIIINTFM

Specifies the name of the NATURAL/IMS interface module to be used by the current driver.

NIMPARM Macro Parameters

The macro NIMPARM generates parameter tables which are contained in the parameter module NIIPARM. The parameters which can be specified with the NIMPARM macro are described below:

ENTRYNM

Possible values:	Any string up to 8 characters.
Default value:	ENV00000

Identifies the current parameter table.

ACTACTV

Possible values:	
YES	An accounting record is written with each terminal I/O.
NO	No accounting record is written.
Default value:	NO

Specifies whether the accounting function is activated.

This parameter is used in dialog-oriented environments only.

ACTAHDR

Possible values:	Any string up to 8 bytes
Default value:	SAG$$$$$

Defines the header of the accounting records if written to the IMS log file. The parameter is only evaluated when the ACTLOG parameter is set to CMD.

This parameter is only used for the accounting function.

ACTARID

Possible values:	
Log code (A0 – FF)	When the ACTLOG parameter is set to LOG.
SMF record type (128 – 255)	When the ACTLOG parameter is set to SMF.
Default value:	None

Specifies the accounting record ID if the accounting record is written using the LOG or SMF settings of the ACTLOG parameter.

This parameter is only used for the accounting function.

ACTLOG

Possible values:	
CMD	Accounting records are written to the IMS log file using the CMD call.
LOG	Accounting records are written to the IMS log file using the LOG call.
SMF	Accounting records are written to SMF using Authorized Services Manager.
Default value:	CMD

Specifies how accounting records are written.

This parameter is only used for the accounting function.

BMPABER

Possible values:	
YES	The BMP run is abended with ADBEND code 3521.
NO	The BMP run is terminated normally with the NATURAL termination error as the condition code. If the BMP run is terminated with a non-recoverable NATURAL/IMS error, condition code 1024 is set.
Default value:	NO

Specifies how a BMP run should be terminated if either a NATURAL runtime error or a NATURAL/IMS interface non-recoverable error occurs.

BROACTV

Possible values:	YES/NO
Default value:	NO

Specifies whether the broadcasting function is available.

This parameter is used in dialog-oriented environments only.

CMBSIZE

Possible values:	Numeric value up to 16 MB.
Default value:	1024

Specifies the size of the command buffer in bytes. The command buffer is used by the service APIs NIICMD and NIIGCMD, the service module CMCMMND and the Accounting function. The size of the command buffer must accommodate the maximum length of the IMS commands to be processed and the maximum length of the accounting record including the user extension.

COLPSCR

Possible values:	Any valid screen width (numeric).
Default value:	80

Specifies the number of columns per screen.

ENDMODN

Possible values:	Any valid MOD name up to 8 characters.
Default value:	DFSMO2

Specifies the MOD name for formatting the screen which appears after a NATURAL session is terminated successfully. This enables NATURAL to be included in a customer-specific menu.

The value of the ENDMODN parameter can be overridden by the service API NIIEMOD and the service module CMEMOD.

Note: If a NATURAL session terminates with an error, DFSMO2 is always used to issue the appropriate NATURAL error message.

ERRLHDR

Possible values:	Any string up to 8 characters.
Default value:	NIIERR$$

Specifies the header of the IMS log records which are written when errors occur in the NATURAL/IMS interface. For further information, see **Recovery Handling** (page 105).

HCBSIZE

Possible values:	Any numeric value up to 16 MB.
Default value:	1024

The size in bytes of the hardcopy print buffer. Records which are sent to a printer destination using the NATURAL hardcopy function are buffered.

HDENSDU

Possible values:	YES/NO
Default value:	NO

Specifies whether a snap dump provoked by a NATURAL/IMS error should be written as a high-density dump to a 3800 printing subsystem.

LINPSCR

Possible values:	Any valid screen size (numeric).
Default value:	24

Defines the number of lines per screen.

MISIZE

Possible values:	Any numeric value up to 16 MB.
Default value:	4096

The size in bytes of the buffer which is to contain the input message. This area must be as large as the largest input message to be received from IMS/TM.

MONACTV

Possible values:	
YES	The session status is written to the SIP server at each terminal I/O.
NO	No session status is maintained.
Default value:	NO

Specifies whether the monitoring function is activated.

This parameter is used in dialog-oriented environments only.

MOSIZE

Possible values:	Any numeric value up to 16 MB.
Default value:	4096

The size in bytes of the buffer which is to contain the output message. This area must be as large as the largest output message to be sent to IMS/TM.

MSACTV

Possible values:	YES/NO
Default value:	NO

Specifies whether the multi-session function is available.

This parameter is used in dialog-oriented environments only.

MSCMPTB

Possible values:	YES/NO
Default value:	NO

Specifies whether sessions should be switched in NII22/NIA-compatible mode.

Only used for the multi-session function.

MSCRKEY

Possible values:	NONE, PF1 – PF24
Default value:	NONE

Specifies with which PF keys a new session can be started.

Note: If MSCMPTB=YES, MSCRKEY must be set to NONE.

Only used for the multi-session function.

MSDBD

Possible values:	Any valid DBD name.
Default value:	None.

Specifies the name of the multi-session database.

Only used for the multi-session function.

MSMAX

Possible values:	2 – 9
Default value:	9

Specifies the highest possible number of parallel NATURAL sessions per terminal.

Only used for the multi-session function.

MSRSKEY

Possible values:	NONE, PF1 – PF24
Default value:	NONE

Specifies the PF key with which an old session can be restarted.

Note: If MSCMPTB=YES, MSRSKEY must be set to NONE.

Only used for the multi-session function.

ROLL SRV

Possible values:	
YES	The NATURAL roll server is used.
NO	Roll files are used.
Default value:	YES

Specifies the medium for saving a NATURAL thread between terminal output and input.

This parameter is used in dialog-oriented environments only.

ROLLFN

Possible values:	1 – 5
Default value:	1

If ROLLSRV=NO, this parameter specifies the number of roll files to be used.

This parameter is used in dialog-oriented environments only.

SPASIZE

Possible values:	Any numeric value up to 16 MB
Default value:	1024

Specifies the size in bytes of the buffer which is to contain the scratch-pad area.

In a non-conversational environment, this is also the size of the simulated SPA which is written to the SIP server.

PRTDRIV

Possible values:	
Drivers for SCS Printers	
SCS–B1	Form feed at start and end of report, starts page on line 1.
SCS–B2	Form feed at start and end of report, starts page on line 2.
SCS–E1	Form feed at end of report, starts page on line 1.
SCS–E2	Form feed at end of report, starts page on line 2.

Possible values:	
SCS–N1	No form feed at start or end of report, starts page on line 1.
SCS–N2	No form feed at start or end of report, starts page on line 2.
SCS–S1	Form feed at start report, starts page on line 1.
SCS–S2	Form feed at start of report, starts page on line 2.
Drivers for non-SCS Printers	
NSCS–B1	Form feed at start and end of report, starts page on line 1.
NSCS–B2	Form feed at start and end of report, starts page on line 2.
NSCS–E1	Form feed at end of report, starts page on line 1.
NSCS–E2	Form feed at end of report, starts page on line 2.
NSCS–N1	No form feed at start or end of report, starts page on line 1.
NSCS–N2	No form feed at start or end of report, starts page on line 2.
NSCS–S1	Form feed at start report, starts page on line 1.
NSCS–S2	Form feed at start of report, starts page on line 2.
Default value:	SCS–S2

Specifies the print driver to be used for reports which are directly written to an IMS/TM printer. For further information, see the section **Support of the NATURAL WRITE(*n*) Statement** on page 40.

SPATID

Possible values:	Any string up to 4 characters
Default value:	NAT2

Specifies the NATURAL subsystem ID for the Authorized Services Manager which is used to save the SPA for a non-conversational driver.

Note: This value must be the same for all parameter tables and must be the same as the value specified for SPATID in the NIMPIXT macro. For further information, see page 61.

For further information about the NATURAL subsystem ID, see the *NATURAL for Mainframes Installation and Operations Manual*.

SUPNONC

Possible values:	YES/NO
Default value:	NO

Specifies whether switching from a terminal-oriented non-conversational environment to a conversational environment is possible.

Only used in the dialog-oriented conversational environment.

TERMDB

Possible values:	YES/NO
Default value:	NO

Specifies whether the NATURAL session has to be terminated if one of the DL/I databases specified in the PSB is not available.

Note: If you set TERMDB to "NO" and one of the databases is not available when it is accessed, the NATURAL transaction code is suspended by IMS/TM.

This parameter is used in dialog-oriented environments only.

TERMIPL

Possible values:	YES/NO
Default value:	NO

Specifies whether a NATURAL session is terminated with an error message when an IPL has taken place between the current transaction step and the start of the session.

This parameter is used in dialog-oriented environments only.

THBELOW

Possible values:	
YES	The NATURAL thread is allocated below the 16 MB line.
NO	The NATURAL thread is allocated above the 16 MB line.
Default value:	YES

Specifies where the NATURAL thread is allocated.

Note: For batch message processing, the thread is always allocated below the 16 MB line.

THSIZE

Possible values:	Any numeric value.
Default value:	300000

Specifies the size of the NATURAL thread. This is the area which contains all user session related NATURAL buffers.

USERID

Possible values:	
YES	The NATURAL user ID specified in *INIT–USER is either taken from the security access control block if a security package is active or from the USER parameter of the job card.
NO	The NATURAL user ID specified in *INIT–USER is taken from the job name.
Default value:	NO

Specifies how the value of the system variable *init–user is determined.

Only used by the BMP driver.

NIMTRNTG Macro Parameters

The macro NIMTRNTG generates an entry in the transaction code table NIITRTAB containing the specified transaction code with related parameters. For each NATURAL transaction code an entry has to be included in the transaction code table. For further information on NIITRTAB, see page 43.

The parameters which can be specified with the macro NIMTRNTG are described below:

TRANCODE

Possible values:	Any valid transaction code name.
Default value:	None.

Specifies the identifier of each entry within the transaction code table.

Note: This parameter has no effect in both the non-message-driven BMP and the batch processing environment.

HCPCB

Possible values:	
SYSPCB	The first alternate TP PCB is used.
WRKPCB	One of the additional alternate TP PCBs is used. This enables you to use an express TP PCB for the hardcopy function.
Default value:	SYSPCB

Specifies which PCB is used for the hardcopy function.

MSGPCB

Possible values:	
SYSPCB	The first alternate TP PCB is used.
OWNPCB	The second alternate TP PCB is reserved and used. This enables you to use an express TP PCB for sending messages.
Default value:	SYSPCB

Specifies which PCB is used when printing error messages and standard output in the non-terminal-oriented environment and for the server driver.

Only relevant for non-terminal-oriented environments and the server driver.

MSPCB

Possible values:	
NO	The multi-session feature is not used
1 – 255	The PCB of the multi-session database
Default value:	NO

Only relevant for the multi-session feature.

NIIPENT

Possible values:	Any non-blank character string up to 8 characters
Default value:	ENV00000

Specifies the name of the NATURAL/IMS parameter table to be used for this entry in the transaction code table.

NRASTART

Possible values:	Any numeric value greater than 14
Default value:	16

Defines the offset of the NATURAL Reserved Area (NRA) within the scratch-pad area.

The current length of the NRA is 157 bytes. The length of the NRA may change from version to version of NATURAL/IMS. If you want to save your own information in the SPA in order to pass it to a non-NATURAL transaction, it is recommended that you save your data in front of the NRA in order to be version compatible.

PSBNAME

Possible values:	Any valid PSB name.
Default value:	None.

Specifies the PSB name corresponding to the current transaction code. This parameter is used to identify the entry in the transaction code table for non-message-driven batch message processing and for the batch processing environment.

TYPE

Possible values:	
CONV	The transaction code is for a conversational NATURAL session.
NONC	The transaction code is for a non-conversational NATURAL session.
Default value:	CONV

Specifies the type of the NATURAL transaction code.

WRKPCBS

Possible values:	
0	No IMS printer is available.
1–32	The number of alternate TP PCBs used for printing *additional* to the first TP PCB and, if appropriate, to the MSGPCB.
Default value:	0

Specifies the number of alternate TP PCBs available for printing *additional* to the first TP PCB and, if appropriate, to the MSGPCB.

Example 1:

You specified the following:

```
MSGPCB=SYSPCB
WRKPCBS=2
```

The PSB must contain 3 alternate TP PCBs.

Example 2:

You specified the following:

```
MSGPCB=OWNPCB
WRKPCBS=2
```

The PSB must contain 4 alternate TP PCBs. The second alternate TP PCB is reserved for the error messages and standard output of the non-terminal-oriented environment.

NIMLPCB Macro Parameters

The macro NIMLPCB can optionally follow a NIMTRNTG entry in the transaction code table.

The parameters which can be specified with the macro NIMLPCB are described below:

NAME

Possible values:	Any non-blank string up to 8 characters
Default value:	None

Specifies the logical name of the PCB.

NUM

Possible values:	Any integer
Default value:	None

Specifies the positional number of the PCB in the PSB. If NUM is not specified, the positional number of the NIMLPCB macro is used.

NIMMSGT Macro Parameters

The macro NIMMSGT generates each entry in the message text module NIIMSGT, which is part of the NATURAL/IMS interface module. Each generated entry provides a message text for each possible NATURAL/IMS error number.

The NIMMSGT macro is specified in one of the following two ways:

N*error-number***[*] NIMMSGT** *message-text*

In this case, NATURAL/IMS will display the message text as defined. The message text may be up to 72 characters long.

X*error-number***[*] NIMMSGT** *message-text*

In this case, NATURAL/IMS will append an error-specific reason code to the current message text. The message text may be up to 64 characters long.

If the error number is followed by an asterisk (*), a snap dump will be generated when an error occurs. You may adapt the message text to your own requirements. You may also add or delete the DUMP option of a specific error number. You must not modify the error number and the characters N or R that precede the error number.

NIMPIXT Macro Parameters

The NIMPIXT macro generates the physical input edit routine. For further information on the physical input edit routine, see page 44.

The parameters which can be specified with the macro NIMPIXT are described below:

SPATID

Possible values:	Any string up to 4 characters
Default value:	none

Specifies the NATURAL subsystem ID for the Authorized Services Manager which is used to save the SPA for the non-conversational driver.

Note: The value of this parameter must be the same as the value specified for the SPATID parameter in the NIMPARM macro.

NIA

Possible values:	YES/NO
Default value:	NO

Specifies whether NIA (NATURAL under IMS/DC Advanced Interface) is supported by the physical input edit routine. If you want to run NATURAL/IMS 2.3 and NIA 2.2 in parallel (on the same terminals), specify YES.

SVC

Possible values:	200 – 255
Default value:	none

Specifies the SVC numbers used by NIA. For more information, see the NATURAL under IMS/DC Advanced Interface documentation.

SVCE

Possible values:	1 – 999
Default value:	200

Specifies the start value for error numbers if errors are detected by the NIA SVC. This value is added to the return code of the NIA SVC to create the error message number.

PIXTE

Possible values:	1 – 999
Default value:	400

Specifies the start value for error numbers if errors are detected by the physical input edit routine. This value is added to the return code set by the physical input edit routine.

SIPSE

Possible values:	1 – 999
Default value:	500

Specifies the start value for error numbers if errors are detected by the Authorized Services Manager. This value is added to the return code set by the Authorized Services Manager.

WTO

Possible values:	YES/NO
Default value:	NO

Specifies whether a WTO message is issued if the Authorized Services Manager fails.

USER

Possible values:	Any string up to 8 characters
Default value:	NO

Specifies whether a user-specific physical input edit routine is to be called if the NIMPIXT macro does not find the SPA. If a user-specific input edit routine is to be called, specify the name of the routine.

The Roll File and Roll Server

These components are used in dialog-oriented environments only.

NATURAL session-related information is held in the NATURAL thread. With each terminal output, the content of the NATURAL thread is saved either in a roll file or by using the roll server. The medium is defined by the NIMPARM parameter ROLLSRV.

To use the roll server, ROLLSRV is set to YES.

Instead of using roll files which have to be allocated to each MPP region, you can use the NATURAL roll server. The roll server offers the following advantages:

- No DD statements in each MPP region.
- One central address space is responsible for access to the roll files.
- Data space support.

Note: In a SYSPLEX environment you must use the roll server.

For further information on roll files and the roll server, see the *NATURAL Installation and Operations Manual for Mainframes*.

To use roll files, ROLLSRV is set to NO.

A roll slot in the roll file is reserved for each NATURAL user.

Roll files are accessed under the DD statements ROLLF1 – ROLLF5.

Note: If your NATURAL transaction code is scheduled in more than one MPP region or if you switch between transaction codes running in different MPP regions, you have to use the same roll files in all MPP regions.

Note: If you reformat the roll file, make sure that no NATURAL transactions are active. If a transaction is scheduled after the roll file has been reinitialized, it cannot locate its roll slot on the roll file and abnormally terminates. To avoid this problem, it is recommended that you cold-start IMS after the roll file has been reformatted.

The Shared NATURAL Nucleus

In an IMS/TM environment, the NATURAL nucleus is always separated from the environment-dependent interface (driver). This means that you have to install the shared NATURAL nucleus. The same NATURAL nucleus can be shared by all NATURAL/IMS environments.

For further information, see the chapter **Operating NATURAL** in the *NATURAL Installation and Operations Manual for Mainframes*.

The NATURAL Buffer Pool

Since NATURAL/IMS is executable in more than one MPP region, it is recommended that the NATURAL buffer pool be a global buffer pool.

Although you can use a local buffer pool, this is not recommended in terminal-driven environments for performance reasons.

For further information on the global buffer pool, see the *NATURAL Installation and Operations Manual for Mainframes*.

The ADABAS Interface

In order to access the NATURAL system file and ADABAS user files, the ADABAS interfaces ADALNK and ADALNI are required. ADALNK is used by the non-terminal-driven environments BMP, NTRD and SRVD. ADALNI is used by terminal-driven environments.

By default, the appropriate ADABAS interface is dynamically loaded at runtime. In dialog-oriented environments, the ADABAS/IMS interface module ADALNI is used. In all other environments, the ADABAS batch interface module ADALNK is used. You can overwrite the name of the ADABAS interface to be used by specifying the NATURAL profile parameter ADANAME.

Note: You must not use the reentrant version of either of these interface modules.

The Preload List

For performance reasons it is recommended that you add the names of the following modules to the preload list for the NATURAL regions:

- the NATURAL/IMS front-ends,
- the NATURAL/IMS interface module,
- the NATURAL shared nucleus,
- the ADABAS interface.

Service Programs

Service programs are NATURAL subprograms which provide NATURAL/IMS with additional functionality. You can call them from within a NATURAL program using a standard CALLNAT statement.

The service programs are provided in the library SYSEXTP and you must copy them to the SYSTEM or steplib library.

Sample NATURAL programs to invoke the service programs are also provided in the library SYSEXTP.

The last parameter in each service program is the return code (I4). The following return code values are common for all service programs:

0	OK
–1	Non-supported function. This is an internal error, please contact SOFTWARE AG support.

If an error occurs, either a NATURAL error message is issued or the session is terminated with a NATURAL/IMS error message.

NIIBRCST

Sends the passed message to the specified terminal using the message output descriptor specified in the Modname parameter.

Parameters

Terminal_name	(A8)
Message	(A1/1:V)
Message_length	(I4)
Modname	(A8)
Return_code	(I4)

Specific Return Code Values

None.

Sample Program

NIPGMSG

NIICMD

Passes the IMS command specified to IMS. If there is a reply, it is moved into the reply area provided. If the reply does not fit into the reply area, it is truncated and the return code is set to 4.

Parameters

Command	(A1/1:V)	Input
Command_length	(I4)	Input
Reply_area	(A1/1:V)	Input/Output
Reply_area_length	(I4)	Input
Reply_length	(I4)	Output
Status_code	(A2)	Output
Return_code	(I4)	Output

Specific Return Code Values

4	Reply truncated

Sample Program

NIPCMD

NIIDEFT

Prepares a deferred switch to the specified NATURAL transaction code. With the next terminal I/O, the output is sent to the terminal and the next input from this terminal is processed by the transaction code specified in the parameter Transaction_code.

Parameters

Transaction_code	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPDEFT

NIIDEFTX

Prepares a deferred switch to a non-NATURAL transaction code. With the next terminal I/O, the output is sent to the terminal using the given modname and the next input from this terminal is processed by the transaction code specified in the parameter Transaction code. If the suspend flag is set to "Y", the NATURAL session will be suspended and can be resumed later. If the NATURAL session is resumed, it will first issue the last NATURAL screen.

Note: If the suspend flag is set to "Y" you may not switch from a conversational NATURAL session to a non-conversational transaction code. If you try to do so, a NATURAL error message is issued.

Parameters

Transaction_code	(A8)	Input
Transaction_type	(A4)	Input Possible values: CONV for conversational NONC for non conversational
Suspend flag	(A1)	Input Possible values: Y the Natural session will be suspended else the NATURAL session will be terminated
Modname	(A8)	Input
Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPDEFTX

NIIDIRT

Prepares a direct switch to a specified NATURAL transaction code. On the next terminal write, the CHNG command to the specified transaction code is issued and the NATURAL screen is inserted using the alternate TP PCB. If you switch from a conversational NATURAL session to a non-conversational one, the conversation is terminated and a dummy message using modname NIIMODNC is inserted. This message unprotects the screen temporarily, and is thus overwritten by the first screen of the non-conversational NATURAL session.

Parameters

Transaction_code	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPDIRT

NIIDIRTX

Prepares a direct switch to the specified transaction code. On the next terminal write, the CHNG call for the new transaction code is issued and the message and or the SPA are inserted using the alternate TP PCB. The transaction type defines the type of the new transaction code. If you switch from a conversational transaction code to a non-conversational one, the conversation is finished by issuing a dummy message using modname NIIMODN, which unprotects the screen temporarily, thus it will be overwritten by the screen issued from the non conversational transaction code. If the suspend flag is set to *"Y"*, the NATURAL session is suspended and may be resumed at a later time. When the NATURAL session is resumed, the last NATURAL screen is issued.

Note: If the suspend flag is set to "Y" and you may not switch from a conversational NATURAL to a non conversational transaction code. If you try to do so, a NATURAL error message will be issued. If message length is set to zero, no message at all is inserted. This however is only possible if you switch to a conversational transaction code.

Parameters

Transaction_code	(A8)	Input
Transaction_type	(A4)	Input Possible values: CONV for conversational transaction code NONC for non-conversational transaction code
Suspend_flag	(A1)	Input Possible values: Y the NATURAL session will be suspended else the NATURAL session will be terminated
Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPDIRTX

NIIEMOD

Modifies the current setting of the module output descriptor to be used in the insertion of the last message in a NATURAL session and sets it to the value specified in the parameter Modname.

Parameters

Modname	(A8)	Input
Return_code	(I4)	Output

NIPEMOD

NIIGCMD

Retrieves the next reply segment of a previously issued IMS/TM command. The length of the reply is return in the parameter reply length. If the reply does not fit into the reply area, the reply is truncated and return code 4 is issued.

Parameters

Reply_area	(A1/1:V)	Input/Output
Reply_area_length	(I4)	Input
Reply_length	(I4)	Output
Status_code	(A2)	Output
Return_code	(I4)	Output

Specific Return Code Values

4	Reply truncated

Sample Program

NIPCMD

NIIGMSG

Retrieves the first segment of the next message from the message queue by issuing a GU. The message area will contain the retrieved message including the leading LLZZ bytes. If there are no messages in the message queue, LLZZ is set to zero.

Parameters

Message_area	(A1/1:V)	Output
Message_area_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPGMSG, NIPGSEG

NIIGSEG

Retrieves the next segment of the input message by issuing a GN call. The message area will contain the retrieved message including the leading LLZZ bytes. If there are no more message segments in the current message, LLZZ is set to zero.

Parameters

Message_area	(A1/1:V)	Output
Message_area_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPGSEG

NIIGSPA

Retrieves data from the SPA beginning at the specified offset in the specified length.

Parameters

Offset	(I4)	Input
Length	(I4)	Input
Area	(A1/1:V)	Input/Output
Return_code	(I4)	Output

Specific Return Code Values

4	The retrieved data resides entirely or partially within the part of the SPA reserved for NATURAL.

Sample Program

NIPGSPA

NIIIMSIN

Retrieves the IMS environment information using the INQY ENVIRON call. If you specify a reply_area_length smaller than 102, the reply will be truncated and you will receive return code X'0100' with reason code X'000C'.

Parameters

Reply_area	(A1/1:v)	Output
Reply_area_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

nnxx	The first two bytes contain the AIB return code. The second two bytes contain the AIB reason code

Sample Program

NIPIMSIN

NIIISRTF

Creates multi-segment messages. NIIISRTF performs the CHNG call for the specified destination and inserts the first message segment without performing a PURG call. Further message segments may be inserted using NIIISRTM (see below). The message has to be terminated using NIIPURG (see below). The LLZZ bytes are created by the service module.

Parameters

Destination	(A8)	Input
Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPISRTM

NIIISRTM

Inserts the next message segment into the message queue without performing a CHNG or a PURG call. The LLZZ bytes are created by the service module.

Parameters

Message	(A1/1:V)	Input
Message_length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPISRTM

NIIPCBAD

Returns the currently scheduled PSB name and the address of the PCB identified by the logical name. If the logical PCB name is not defined in the transaction code table, a NATURAL error message is issued.

Parameters

PSB_name	(A8)	Output
Logical_PCB_name	(A8)	Input
PCB_address	(B4)	Output
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPPCBAD

NIIPCOM

Moves the data provided in the data area into the reply area specified in the NIIBOOTS call at the specified offset in the specified length. NIIPCOM may be called from the server environment only .

Parameters

Offset	(I4)	Input
Data_area	(A1/1:V)	Input
Length	(I4)	Input
Return_code	(I4)	Output

Specific Return Code Values

4	Calling environment not server environment

Sample Program

NIPPCOM

NIIPMSG

Sends a message using a given MOD name to the destination which is represented by the I/O PCB. The message is taken from the message area in the specified message area length. The message area must not contain the leading LLZZ bytes. In this way you can send MFS-formatted output messages back to the originator of the input message.

Parameters

Message	(A1/1:V)	Input
Message_length	(I4)	Input
Modname	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPPMSG

NIIPSBAD

Returns the address of the PSB, which is the address of the PCB address list.

Parameters

PSB_address	(B4)	Output
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPBOOTS

NIIPSPA

Replaces the data located in the SPA at the specified offset in the given length by the data provided in the data area.

Parameters

Offset	(I4)	Input
Length	(I4)	Input
Data_area	(A1/1:V)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Note: An attempt to override the header of the SPA (first 14 bytes) and/or data residing in the NATURAL-reserved area is refused and a NATURAL error message is issued.

Sample Program

NIPPSPA

NIIPURG

Issues a PURG call.

Parameters

Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPISRTM

NIIRETRM

Moves data from the input message beginning at the specified offset in the specified length into the provided message area.

Note: The offset is calculated from the LLZZ bytes.

Parameters

Offset	(I4)	Input
Length	(I4)	Input
Message_area	(A1/1:V)	Input/Output
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPRETRM

NIISASD

Modifies the current setting of the NATURAL dynamic parameters SENDER and OUTDEST.

Parameters

Sender	(A8)	Input
Outdest	(A8)	Input
Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPNTRD

NIIU3962

Terminates the session with user abend 3962 and produces a dump.

Parameters

Return_code	(I4)	Output

Specific Return Code Values

None.

Sample Program

NIPU3962

Service Modules

These modules perform IMS/TM-specific functions. They can be called from within a NATURAL program using the standard NATURAL CALL interface. Sample programs are loaded by a NATURAL INPL into the library SYSEXTP.

This section contains a detailed description of all the service modules in alphabetical order.

CMCMMND

The module CMCMMND issues IMS operator commands and returns the reply segments to the NATURAL user program.

Parameters

Command	Input
Command length (B4)	Input
Reply	Output
Length of reply area (B4)	Input

The operator command contained in the command area is issued to IMS with the indicated length. If the user has set a non-zero reply length, any reply segments from IMS are moved into the reply area over the maximum available length. If the reply area is at least two bytes long, the first two bytes contain the IMS status code after the command call has been issued. The two rightmost bytes of the REPLGTH field contain the effective length of the total reply moved into the REPLY field. If the reply from IMS has to be truncated, this is indicated by setting X’80’ in the leftmost byte of the REPLGTH field.

Sample Program

NIPSCMND

CMDEFSW

The module CMDEFSW performs a deferred transaction switch to a NATURAL transaction code.

Parameters

Trancode	Input

With the next terminal I/O, the output is sent to the terminal and the next input from this terminal is processed by the transaction code passed as parameter message.

CMDEFSWX

The module CMDEFSWX performs a deferred switch to a non-NATURAL transaction code.

Parameters

Trancode	Input
Message	Input
Message length	Input
Modname	Input

With the next terminal I/O, the given message with the given module name is inserted and the NATURAL session is terminated.

If the new transaction code is a NATURAL transaction code, the message and the module name passed as parameters are ignored and the “real” message is inserted using MFS bypass.

Sample Programs

NIPSDEFX.

CMDIRNMX

The module CMDIRNMX has the same functionality as CMDIRSWX, except that no message is inserted to the alternate PCB. Thus, the only parameter you have to provide is TRANCODE.

Parameters

Trancode	Input

CMDIRNMX can also be used to perform a direct switch to another NATURAL transaction code, because in this case, the CLEAR key is given as input message to NATURAL by default.

CMDIRNMZ

The module CMDIRNMZ has the same functionality as CMDIRSWZ, except that no message is inserted to the alternate PCB. Thus, the only parameter you have to provide is TRANCODE.

Parameters

Trancode	Input

CMDIRSWX

The module CMDIRSWX performs a direct switch to another conversational transaction and specifies a message that is to be passed on to this new transaction.

Parameters

Trancode	Input
Message	Input
Message length (B4)	Input

At the next terminal I/O, a change call is executed against the alternate PCB to set its destination to the value of the TRANCODE field. The SPA and the message are then inserted into the alternate PCB.

The new transaction code is checked if it is a NATURAL or a non-NATURAL transaction code.

In the case of a non-NATURAL transaction code, the NATURAL session is terminated.

In the case of a NATURAL transaction code, the CLEAR key is passed to NATURAL as input message, which means that NATURAL reacts as if the terminal user pressed the CLEAR key. The type of the new transaction code is automatically honored.

Sample Programs

NIPSDIRX.

CMDIRSWZ

The module CMDIRSWZ has the same functionality as CMDIRSWX.

Parameters

Trancode	Input
Message	Input
Message length (B4)	Input

The difference compared to CMDIRSWX is that in byte NRAOLDST+3, an additional flag is turned on to indicate that the current roll buffer must be given to the swap pool manager, even if a switch to a non-NATURAL transaction code is performed. This is done with the following intention:

- A given NATURAL session gives control to a non-NATURAL transaction code; the session is not terminated.
- The non-NATURAL transaction performs a terminal I/O and then switches back to the original NATURAL transaction, passing data into the SPA.
- The NATURAL transaction does not start a new session, but continues the old session were it has left it, which means that the roll slot is obtained from the swap pool and control is given to NATURAL so as to continue with an existing session.

The non-NATURAL transaction code must pass the message “LLZZD”, where “LL=H’0005’”, “ZZ=X’0000’” and “D=X’6D’” are simulating to NATURAL that CLEAR has been pressed. By making the NATURAL program sensitive to the CLEAR key, it is able to recognize that the called non-NATURAL transaction has come back and it can retrieve the data prepared by the non-NATURAL transaction for use in subsequent processing.

Note: CMDIRSWZ cannot be used if the transaction code to switch to is a NATURAL transaction code.

Sample Program

NIPSDIFS

CMDISPCB

The module CMDISPCB is used to obtain the contents of a PCB.

Parameters

PCB number (B4)	Input
Receiving area	Output
Area length (B4)	Input

After the call is executed, the receiving area contains the contents of the PCB with the requested number in the requested length. A check is made to verify that the requested PCB is within your current PCB list. The first PCB is PCB number 1, the second PCB is PCB number 2, etc.. If you specify an invalid number, the field PCBNUM is set to X'FFFFFFFF' and no further information is passed to your application program.

Sample Program

NIPSPCBD

CMEMOD

The module CMEMOD allows the module name to be modified dynamically for a given LTERM at the normal end of a NATURAL session.

Parameters

Modname (A8)	Input

At a normal end of a session, the environment-dependent interface inserts the message X'00060000403F' into the IOPCB, using the module name whose value is contained in Modname parameter. This is intended to present a meaningful screen (for example, a general menu) to the terminal user so that he can continue working at the terminal.

CMGETMSG

The module CMGETMSG reads the next message from the message queue.

Parameters

Message area	Output
Message area length (B4)	Input

The length is checked to see if the received message fits into the message area. The message is moved including the LLZZ bytes into the message area. If there are no more messages, LL=0 is moved into the message area.

If the message does not fit into the message area, a corresponding error message is returned.

Sample Programs

NIPSGETM and NIPSOBMP.

CMGETSEG

The module CMGETSEG reads the next segment of the current message from the message queue.

Parameters

Message area	Output
Message area length (B4)	Input

The length is checked to see if the received message fits into the message area. The message segment is moved including the LLZZ bytes into the message area. If there are no more message segments, LL=0 is moved into the message area.

If the message does not fit into the message area, a corresponding error message is returned.

Sample Program

NIPSOBMP

CMGETSPA

The module CMGETSPA transfers the data from the SPA starting from the given offset in the requested length into the receiving area.

Parameters

Offset (B4)	Input
Length (B4)	Input
Area (B4)	Output

Sample Programs

NIPSGSPA and NIPSPSPA

CMIMSID

The module CMIMSID enables NATURAL programs to obtain the MVS subsystem ID of the IMS system in which they are currently scheduled.

Parameters

IMSID (A4)	Output

After the call is executed, the field IMSID contains the MVS subsystem ID of the IMS system in which you are currently scheduled.

The module CMIMSID depends upon an internal IMS control block. Therefore, it is an IMS release-dependent function that will be updated whenever possible.

CMIMSINF

The module CMIMSINF provides system environment information.

Parameters

IMSID (A4)	Output. The IMS ID
SUFFIX (A2)	Output. The preload suffix
APPLGNAM (A8)	Output. The application group name
APPLNAM (A8)	Output. The application name
NRENT (B4)	Output. The number of reentrant modules preloaded
NNONR (B4)	Output. The number of non-reentrant modules preloaded

CMIMSINF is also an IMS release-dependent module.

Sample Program

NIPSINF

CMPCBADR

The module CMPCBADR returns the address of a PCB which is identified by a logical name. The PSB name is also returned to the NATURAL program.

Parameters

PSBNAME (A8)	Input
PCBNAME (A8)	Input
PCBADR (B4)	Input

After the call is executed, the field PCBADR contains the address of the PCB identified in the table module by the logical name “PCBNAME” in the table entry that corresponds to the currently scheduled transaction code. If the logical name does not exist for this transaction code, X’FFFFFFFF’ is returned in the PCBADR field. In any case, the field PSBNAME contains the name of the currently scheduled PSB.

Sample Program

NIPSPCBA

CMPRNTR

The module CMPRNTR changes the default hardcopy destination set by the module NIIIMSHC to the value passed as parameter.

Parameters

Destination (A8)	Input

The module CMPRNTR is provided for compatibility reasons only; use the NATURAL SET CONTROL *hdest-id* statement instead.

CMPUTMSG

The module CMPUTMSG can be used to insert any given output message of a given length into the IO-PCB, using any given MFS module name.

CMPUTMSG takes the number of bytes as indicated in the message length from the message area and inserts them with the specified module name in the message queue. There is no restriction upon the length of the message, except that it has to fit into the input message area of the environment-dependent interface. No check is made regarding the IO-PCB status code after the insert call is issued to IMS/TM. In this way, you can send MFS-formatted output messages back to the originator of the input message.

Parameters

Message area	Input
Message length (B4)	Input
Modname	Input

CMPUTSPA

The module CMPUTSPA moves the data with the given length at the specified offset into the SPA.

Parameters

Offset (B4)	Input
Length (B4)	Input
Data	Input

A check is done if the specified offset points into the NATURAL Reserved Area (NRA) within the SPA. If yes, return code 4 is returned.

Sample Program

NIPSPSPA

CMQTRAN

The module CMQTRAN returns the contents of the current entry within the transaction code table.

Parameters

Transaction code	Output. The transaction code under which you are running.
Offset (B2)	Output. The offset of the NRA with the SPA.
Length (B2)	Output. The length of the NRA.
Uoffset (B2)	Output. Not used.
PSB name	Output. The name of the scheduled PSB.
Number of PCBs	Output. The number of PCBs whose addresses you can obtain using the module CMPCBADR.

The logical names by which you can refer to PCBs in the module CMPCBADR are not returned because of security considerations; you should be informed by your system about which logical names you are allowed to refer to.

Sample Programs

NIPSQTRA

CMQUEUE

The module CMQUEUE inserts a message into the first alternate PCB.

Parameters

Destination	Input
Message	Input
Message length (B4)	Input

This call causes an immediate change call to set the destination of the first alternate PCB to the value contained in the field DESTINATION, after which the message is inserted into the alternate PCB with the indicated length (contained in the field MSGLGTH).

The transaction code is inserted after the LLZZ bytes with a length of 8.

After a PURGE call has been issued, control is returned to the next instruction in the NATURAL program.

The message can have any length up to the size of the input message area (usually 8000 minus 12 bytes).

Sample Program

NIPSQLOA

CMQUEUEX

The module CMQUEUEX provides you with complete control over the contents of a message that is to be queued in the IMS/TM input queue.

Parameters

Destination	Input
Message	Input
Message length (B4)	Input

This call causes an immediate change call to set the destination of the first alternate PCB to the value contained in the field DESTINATION, after which the message is inserted into the alternate PCB with the indicated length (contained in the field MSGLGTH) after the LLZZ bytes. The difference compared to CMQUEUE is that the transaction code is *not* inserted after the LLZZ bytes.

After a PURGE call has been issued, control is returned to the next instruction in the NATURAL program. The message can have any length up to the size of the input message area (usually 8000 minus 12 bytes).

Sample Program

NIPSQUEX

CMSNFPRT

The module CMSNFPRT sets the logical name of the device to which the NATURAL messages during the online BMP run is sent.

Parameters

Printer name	Input

Note: Before calling CMSNFPRT, use the NATURAL parameter SENDER to define the default output destination.

Sample Program

NIPSOBMP

CMSVC13D

The module CMSVC13D terminates the NATURAL session with user abend 222 and creates a dump.

Parameters

None

Sample Program

None.

CMTRNSET

When the NATURAL session is terminated normally, the NATURAL/IMS interface performs a direct program-to-program switch to the specified transaction code and inserts the SPA via the alternate PCB.

Parameters

Trancode	Input

Note: The module CMTRNSET is available only if the multi-session feature is implemented.

Sample Program

NIPSEOSS

NIIDDEFS

Module NIIDDEFS is similar to module CMDEFSWX. If you use NIIDDEFS to perform a deferred switch to a foreign transaction, the current NATURAL session is suspended, as with module CMDIRSWZ. The suspended NATURAL session can be resumed at any time by sending back to NATURAL a message containing the CLEAR key.

Parameters

Transaction code	Input. The transaction code to switch to.
Message	Input. The message to be sent to the foreign transaction code.
Message length (B4)	Input
Module name (A8)	Input
Transaction Type (A4)	Input. An A4 variable containing the string “CONV” if the foreign transaction is conversational and the string “NONC” if the foreign transaction is non-conversational.

Return Codes

0	OK
4	The message length is greater than the size of the message area defined in the environment table.
8	You tried to do a deferred switch with suspend from a conversational NATURAL to a non-conversational foreign transaction, something which cannot be done.
12	The fifth parameter is invalid; it contains neither “CONV” nor “NONC”.

Sample Program in Library SYSTP

NIPSDEFS

NIIDPURG

The module NIIDPURG does not have parameters. It issues a PURG call using the first alternate PCB and inserts multi-segment messages using the module NIIDQUMS.

Return Codes

Either bytes two and three of the 4-byte return code contain the status code, or the return code has the value 0.

Sample Program in Library SYSTP

NIPSQLMS

NIIDQUMS

This module creates multi-segment messages. It has basically the same functionality as the module CMQUEUE, with the difference that NIIDQUMS does not issue a PURG call.

Parameters

Destination	Input
Message	Input
Message length (B4)	Input

Note: It is your responsibility to issue the PURG call using the module NIIDPURG.

Sample Program in Library SYSTP

NIPSQLMS

NIIDSETT

In order to perform a correct transaction switch to a foreign transaction code, the type of the foreign transaction code must be known. To obtain this type, the special-purpose module NIIDSETT can be used. If NIIDSETT is not used, the foreign transaction code is assumed to be of the same type as the invoking NATURAL transaction code. If this is not the case, there will be unpredictable results or the session will terminate abnormally.

Parameters

Transaction type (A4)	Input. Possible values: “CONV” for conversational, “NONC” for non-conversational.

Special Functions

This section describes the use of special functions available with the NATURAL/IMS interface and covers the following topics:

- Accounting (page 94)
- Monitoring (page 95)
- Broadcasting (page 97)
- The multi-session feature (page 98)
- The server environment (page 100)

Some of these functions require the Authorized Services Manager (ASM). If the ASM is required, it must have been started before the function is used. The NATURAL subsystem used by the ASM must be the same as the one used by the NATURAL session. For accounting and monitoring, the SIP server must have been enabled in addition.

Accounting

The accounting function is only available in dialog-oriented environments. It is activated by setting the environment table parameter ACTACTV to “YES”.

With each terminal I/O, information about the specific NATURAL session is written to the IMS log or to SMF, depending on the setting of the NATURAL/IMS parameter ACTLOG.

If the ACTLOG parameter is set to “CMD”, a /LOG command is issued that writes the accounting record to the IMS/TM log. All transaction codes must therefore be allowed to use the /LOG command.

At the beginning of each record an 8-byte header is inserted. This header helps to easily select the accounting records using the IMS utility DFSERA10. The header string is defined by the environment table parameter ACTAHDR.

If the ACTLOG parameter is set to “LOG”, the accounting record is written to the IMS/TM log using the LOG call. With the NATURAL/IMS parameter ACTARID, you specify the log code to be used.

If the ACTLOG parameter is set to “SMF”, the accounting record is written to SMF using the Authorized Services Manager. With the NATURAL/IMS parameter ACTARID, you specify the SMF record type to be used.

Monitoring

The monitoring function is only available in dialog-oriented environments. It is activated by setting the environment table parameter MONACTV to “YES” and uses the SIP function of the Authorized Services Manager. The NATURAL subsystem must be the same as the one used by the NATURAL session to be monitored. You can follow the ongoing activity of all NATURAL sessions which use the same NATURAL subsystem by using the Monitoring (M) function of the SYSTP utility. For more information on the SYSTP utility, see the *NATURAL Utilities Manual for Mainframes*. The SYSTP session must also use the same NATURAL subsystem.

The following information about each NATURAL user session is stored with each terminal I/O:

- IMS ID of the IMS system in which the user is active,
- LTERM name of the IMS terminal on which the session was started,
- user ID of the user of the NATURAL session (taken from the IOPCB),
- number of dialog steps currently performed,
- currently active transaction code,
- currently active PSB name,
- current NATURAL library name to which the user is logged on,
- currently active NATURAL program name,
- non-NATURAL transaction code to which the session is possibly suspended to,
- time and date when the session was started,
- time and date of the last ENTER operation,
- DBID and FNR of the NATURAL system file (FNAT) for this session,
- DBID and FNR of the NATURAL user file (FUSER) for this session,
- DBID and FNR of the NATURAL dictionary file (FDIC) for this session,
- DBID and FNR of the NATURAL SECURITY system file (FSEC) for this session,
- DBID and FNR of the NATURAL spool file (FSPOOL) for this session,
- DBID and FNR of the SUPER NATURAL system file for this session,
- last encountered NATURAL error number,
- compressed thread length of the last terminal output.

The information is mapped by the DSECT NIMACTR. There are two areas for storing the DBID and FNR of the NATURAL system files used. In the first area, one byte is used for each DBID and FNR; this is supported for compatibility reasons. In the second area, a fullword is used for each DBID and FNR to support ADABAS Version 6. The accounting record is prefixed with a length and version field.

Before the accounting record is written to the IMS/TM log, respectively to SMF, the user exit NIIXACCT is called. You can use this user exit to tailor the accounting record to your requirements. You may also append information to the accounting record. In this case, you must set the length field to the new length.

Since the accounting record is built in the command buffer, the total length must not exceed the value specified with the NATURAL/IMS parameter CMBSIZE minus 17 bytes. The maximum length allowed is passed as parameter.

If NIIXACCT returns with a non-zero value in register 15, no accounting record is written.

Broadcasting

The broadcasting function is only available in dialog-oriented environments. It is activated by setting the environment table parameter BROACTV to “YES” and uses the SIP function of the Authorized Services Manager.

Once broadcasting is active, it is possible to send broadcast messages to targeted users of a given NATURAL subsystem. Such users can be:

- all users of the NATURAL subsystem to which the sender is connected;
- all users of the NATURAL subsystem within the same IMS system as the sender of the message;
- all users of the NATURAL subsystem within the same IMS system as the sender of the message, but additionally restricted to a given transaction code;
- all users of the NATURAL subsystem within the same IMS system as the sender of the message, but additionally restricted to a NATURAL application;
- all users of the NATURAL subsystem within the same IMS system as the sender of the message, but additionally restricted to a NATURAL application and to a given FUSER system file.

When a session comes to a terminal output, a check is made to see whether the session has to receive a message or not. If not, the “normal” NATURAL output is sent. If yes, the message is sent instead of the normal output and, when pressing ENTER, the NATURAL nucleus is instructed to re-send the last screen. In this way, you first see the message and afterwards receive the “normal” screen. If you want to send more than one message, you have to send them one after the other until the last message has been sent. Afterwards, the normal screen is displayed. When a broadcast message is sent, you must press RESET before you can press ENTER again. All possible attention IDs have the same effect as pressing ENTER.

The application SYSTP can be used to create broadcast messages and to display the contents of all active messages together with the LTERM/IMSID of the sender. The text of a message is limited to 72 bytes.

Messages to be broadcast are saved in a pool maintained by the SIP server. They remain there until you delete them using the SYSTP utility or until you shut down the Authorized Services Manager.

Note: When a broadcast message is deleted or created, all expired messages are deleted as well.

The Multi-Session Feature

Functionality

The multi-session feature is only available in dialog-oriented environments. It is activated by setting the NATURAL/IMS parameter MSACTV to “YES” and allows to run up to nine simultaneous NATURAL sessions on the same logical terminal.

With NATURAL/IMS 2.3, new sessions are created and suspended sessions are resumed in a different way as with NATURAL/IMS 2.2. The NATURAL/IMS 2.2 mode is still supported for compatibility reasons, but is no longer documented. The old 2.2 mode is activated by setting the NATURAL/IMS parameter MSCMPTB to “YES”.

With NATURAL/IMS 2.3, creating and resuming NATURAL sessions is controlled using PF keys. The NATURAL/IMS parameters MSCRKEY and MSRSKEY define the create and resume keys.

If the create key is pressed, the current NATURAL session is suspended and a new NATURAL session is created.

If the resume key is pressed, any input field may contain the string *x*, where $1 \leq x \leq$ MSSESMX. In this case, the currently active session is suspended and the specified session is resumed. If *x* is greater than MSSESMX, the input is passed to the active application.

If the resume key is entered without any input, the next suspended session is resumed in a wrap-around manner. If, for example, the active session is session number 4, the next suspended session in the range 5 through MSSESMX or 1 through 3 is resumed. If there is no suspended session, the input is passed to the active application.

If the session to be resumed is using a different transaction code than the current session, an implicit program-to-program switch is done in order to resume the suspended session with the correct transaction code.

If conversational and non-conversational environments are used in parallel, the modnames NIIMODNC and NIIMODMS are used for switching between the environments. The format definitions are delivered in the source modules NIIMODMS and NIIMODNC and have to be defined to MFS using MFSUTL.

On session termination, the active session is terminated and the next suspended session (if there is any) is resumed in the same order as with the resume key.

If the NATURAL/IMS parameter MSMAX is set to “2” (two parallel sessions are allowed), the create key must be identical with the resume key. In this case, pressing the create key creates a second session if only one session is active, if not, it switches the session.

The Session ID

The session ID is the internally used unique identification of a session. It has the form *XXXXXXxY*, where *XXXXXXx* is the prefix and *Y* is the session number in the form of a numeric digit in the range of 1 to 9.

The session prefix is built by compressing the logical terminal name into a 6-byte binary number *XXXXXX* and by setting *x* to binary zero.

If the compression algorithm is not suitable for the LTERM names used (error 3635 is issued in this case), the user exit NIIXMSSP must be used in order to build a unique 7-byte prefix of the session ID.

The Multi-Session Database

The multi-session database is a HDAM root-only database which contains the NATURAL-reserved area of the SPA for each suspended session.

The model DBD for the description of the multi-session database is delivered in the source module NIIMSDBD and must be defined to IMS/TM.

The DBD name must be specified in the NATURAL/IMS parameter module using the parameter MSDBD. The PCB number must be specified in the transaction code table using the parameter MSPCB of macro NIMTRNTG.

The Server Environment

The server environment allows 3GL applications to execute NATURAL programs using a call interface. It is available in all supported IMS/TM environments and consists of a special NATURAL/IMS driver, NIISRVD, of the call interface NIIBOOTS and of the service API NIIPCOM.

NIISRVD and NIIBOOTS are delivered as source modules and must be assembled and link-edited on your site. For details, see the section **Installing the NATURAL/IMS Interface**.

The server environment allows you to start a NATURAL session by calling NIIBOOTS from any 3GL program. After the NATURAL session has been started, it returns to the calling 3GL program and waits for further input. The input would normally be expected from CMSYNIN, which means that the 3GL program has to simulate NATURAL's primary input dataset.

It is strongly recommended to always put the server NATURAL on the NEXT line. This allows the next call to NIIBOOTS to either execute a NATURAL command or a NATURAL program. Otherwise, the next call to NIIBOOTS would be treated as input for a NATURAL program which had been started by a previous call to NIIBOOTS.

Similarly as with the message-oriented interface, all output normally written to CMPRINT is sent to the IMS/TM destination specified with the NATURAL profile parameter SENDER.

Note: In an MPP environment, the same server NATURAL will be used by all transactions scheduled in this region.

The Call Interface NIIBOOTS

NIIBOOTS is the default name as used in the documentation and in the delivered sample programs. This default name can be changed during installation.

NIIBOOTS requires the following parameters:

- the PSB address (the address of the PCB address list),
- the command area,
- the reply area.

In the command area, the following may be passed:

- the startup parameters,
- any NATURAL command followed by its input data,
- the NIIBOOTS-specific commands, such as STAT and REFR (in combination with the startup parameters).

The startup parameters are passed in two contiguous 80-byte areas. The first area contains the name of the environment table to be used as follows:

ENV-TAB=*environment-table-name*

The second area contains the dynamic NATURAL parameters with which the NATURAL session is to be started.

The reply area is the area in which a reply is to be entered from the executed NATURAL program using the service API NIIPCOM.

Each time it is invoked, NIIBOOTS checks whether the server NATURAL has been initialized.

- If NATURAL has not been initialized, a new NATURAL session is started and the received command is passed to NATURAL as a dynamic parameter.
- If NATURAL has been initialized, the string received in the command area is passed to NATURAL as a NATURAL command or as a NATURAL program.

The NIIBOOTS-specific commands STAT and REFR do the following:

- STAT returns “COLD” in the reply area if NATURAL has not been initialized and “WARM” if it has been initialized.
- REFR forces the initialization/reinitialization of NATURAL, regardless of the current state of NATURAL.

ON ERROR Routine Recommended

It is highly recommended to use an ON ERROR routine in the executed NATURAL programs in order to give back to the calling 3GL program some information in the reply area using NIIPCOM.

Return Codes

NIIBOOTS passes the return code provided by NATURAL on the termination of NATURAL.

Sample Programs

To illustrate usage of NIIBOOTS and NIIPCON, the sample programs NIPBOOTS and NIPPCOM are provided. NIPBOOTS plays the role of the calling 3GL program, NIPPCOM is a sample NATURAL program executed in the server environment and writes the string “NIISRVR” into the reply area. The ON ERROR routine places the NATURAL error number in the reply area.

With the sample programs, you can go through the following scenario:

1. Pass the command “STAT”. The string “COLD” is returned to the reply area.
2. Pass the command: **STACK=(LOGON SYSEXTP),SENDER=S0201**
   “S0201” is the LTERM name of the assigned printer device in the server NATURAL.
   NATURAL will be initialized and will be ready to receive a NATURAL command in library SYSEXTP.
   The successful logon message is issued on the assigned printer. Nothing is returned in the reply area.
3. Pass the command “STAT”. The string “WARM” is returned to the reply area.
4. Pass the command “NIPPCOM”. Program NIPPCOM is executed and the string “NIPSRVR” is returned to the reply area. NATURAL is ready to accept the next command in library SYSEXTP.
5. Pass the command: **REFR STACK=(LOGON SYSEXTP;NIPPCOM),SENDER=S0201**
   NATURAL is reinitialized and program NIPPCOM in library SYSEXTP is executed. The reply area contains the string “NIPSRVR”.
6. Pass the command “FIN”.
   NATURAL is terminated and no information is passed to the reply area. The return code will contain the return code of the NATURAL termination. The NATURAL termination message is issued on the assigned printer device.
7. Pass the command “STAT”. The string “COLD” is returned to the reply area.

User Exits

This section describes the user exits available with the NATURAL/IMS interface. It contains an alphabetical list of the user exits available with NATURAL for IMS/TM. For each exit, a source module with the same name is provided. Each source module contains a description of the parameter list and of the register conventions.

NIIXSTAR

The NIIXSTAR routine is called with each transaction step after the SPA and the message have been retrieved. Within this module, the NATURAL reserved area within the SPA, the NATURAL terminal I/O control block and the input message are available.

Note: At this point, the IOCB is not yet initialized.

NIIXMSSP

The NIIXMSSP module is only called if the multi-session feature is in use. It builds the first 7 bytes of the session identification. By default, NATURAL/IMS compresses the LTERM names to 7 characters.

NIIXSSTA

The NIIXSSTA module is called when a new NATURAL user session is initialized. A value of 12 in register 15 upon return of NIIXSSTA forces the NATURAL/IMS interface to terminate the NATURAL session. Any other non-zero value in register 15 forces the interface to issue the NATURAL/IMS interface error 3509 with the reason code containing the value in register 15. When entering the module NIIXSSTA, register 10 points to the initialized terminal I/O control block (IOCB), from which it obtains all the necessary information. The IOCB is mapped by the macros NAMIOCB and NIMNRA.

NIIXISRM

The NIIXISRM module is called before the insertion of the message into the IOPCB.

NIIXISRT

The NIIXISRT module is called before the insertion of the SPA into the IOPCB, even at the end of the NATURAL session. The end-of-session situation can be recognized by a blank transaction code.

NIIXACCT

The NIIXACCT module receives control before an accounting record is written to the IMS log or to SMF. Thus, it makes it possible to modify the content of an accounting record. If NIIXACCT returns a non-zero register 15, the accounting record is not written at all.

NIIXTGN0

The NIIXTGN0 module is called when the service module CMGSEGO or CMGETSEG is used. NIIXTGN0 receives control immediately after the message segment is retrieved, regardless of the status code.

NIIXTGU0

The NIIXTGU0 module is called when the service module CMGETMSG is used. NIIXTGU0 receives control immediately after the GU call against the IOPCB, regardless of the status code.

NIIXJESA

The NIIXJESA module is called when the JES API is used for writing reports. It is called after the options string has been created and may be used to modify the options string.

NIIXPRT0

The NIIXPRT0 module is called when reports are directly written to IMS/TM printers. It can be used to set the codes for “form feed” and “new line”.

NIIXRFNU

The NIIXRFNU module is called when the new NATURAL session is assigned to a roll file. It can be used to calculate the number of the roll file to be used for this session.

Recovery Handling

This section describes recovery handling in the NATURAL/IMS interface and covers the following topics:

- Abends (page 105)
- Non-Recoverable Errors (page 105)
- Recoverable Errors (page 105)

Abends (System and User Abends)

The NATURAL/IMS interface is protected by an ESTAEX environment which takes control in case of an abend. If a user abend is detected, resources are cleaned up and the abend is percolated without giving control to NATURAL. If a system abend is detected, NATURAL is informed about the abend and depending on the setting of the NATURAL profile parameter DU, NATURAL contines with an error message or terminates the session. In both cases, you can produce a dump which represents the situation at the time when the error occurred (register contents, PSW, etc.). The dump is produced if DU=ON or if the user abend has requested this.

Non-Recoverable Errors

A non-recoverable error is a logical error detected by the NATURAL/IMS interface which cannot be handled by NATURAL. These situations typically occur during startup, termination or terminal I/O. In all cases, the NATURAL runtime is not active and can thus not react to the error.

If a non-recoverable error is detected, the NATURAL/IMS interface issues a NII error and terminates the session. The error message is also written to the IMS log and to the system log. Depending on the dump option in the error message table, a snap dump is produced.

Note: If it is not possible to send the error message (for example if the GU has failed), the session abends (user abend).

Recoverable Errors

If a logical error is detected by the NATURAL/IMS interface which can be handled by NATURAL, for example an invalid destionation for a report, a NATURAL error message is issued and NATURAL proceeds with its standard error handling.

Installing the NATURAL/IMS Interface

This section describes how to install the NATURAL/IMS interface (NII) and covers the following topics:

- Prerequisites (page 106)
- The Installation Tape (page 107)
- Installing the NATURAL/IMS Interface (page 108)
- Installing the Multi-Session Feature (page 123)
- Installation Verification (page 124)

Prerequisites

The following software must be installed and running before you install the NATURAL/IMS interface:

- Base NATURAL Version 2.3 under MVS.
  For further information, see the *NATURAL Installation and Operations Manual for Mainframes*.
- NATURAL global buffer pool if you are using the MPP environment (strongly recommended).
- NATURAL roll server if the NATURAL/IMS parameter ROLLSRV is set to YES.
- Authorized Services Manager with the SIP Server function if Monitoring or Broadcasting is required.
- Authorized Services Manager if Accounting to SMF is required.
- ADABAS/IMS interface.
- IMS/TM.

The Installation Tape

The installation tape contains the datasets listed in the table below. The sequence of the datasets is shown in the *Report of Tape Creation* which accompanies the installation tape.

Dataset Name	Contents
NII*nnn*.LOAD	IMS-dependent load modules
NII*nnn*.SRCE	IMS-dependent source programs and macros

The notation *nnn* in dataset names represents the version number of the product.

Copying the Tape Contents to Disk

If you are not using SYSTEM MAINTENANCE AID, adapt and run job NIITAPE to copy the load and source libraries from tape to disk. NIITAPE is contained in job dataset NAT*nnn*.JOBS on the NATURAL installation tape. The sample jobs use the sequential datasets directly from tape.

The dataset type and disk space requirements are shown in the *Report of Tape Creation*.

Sample Jobs

The sample jobs are contained in the dataset NAT*nnn*.JOBS and are prefixed with “NII”.

Installing the NATURAL/IMS Interface

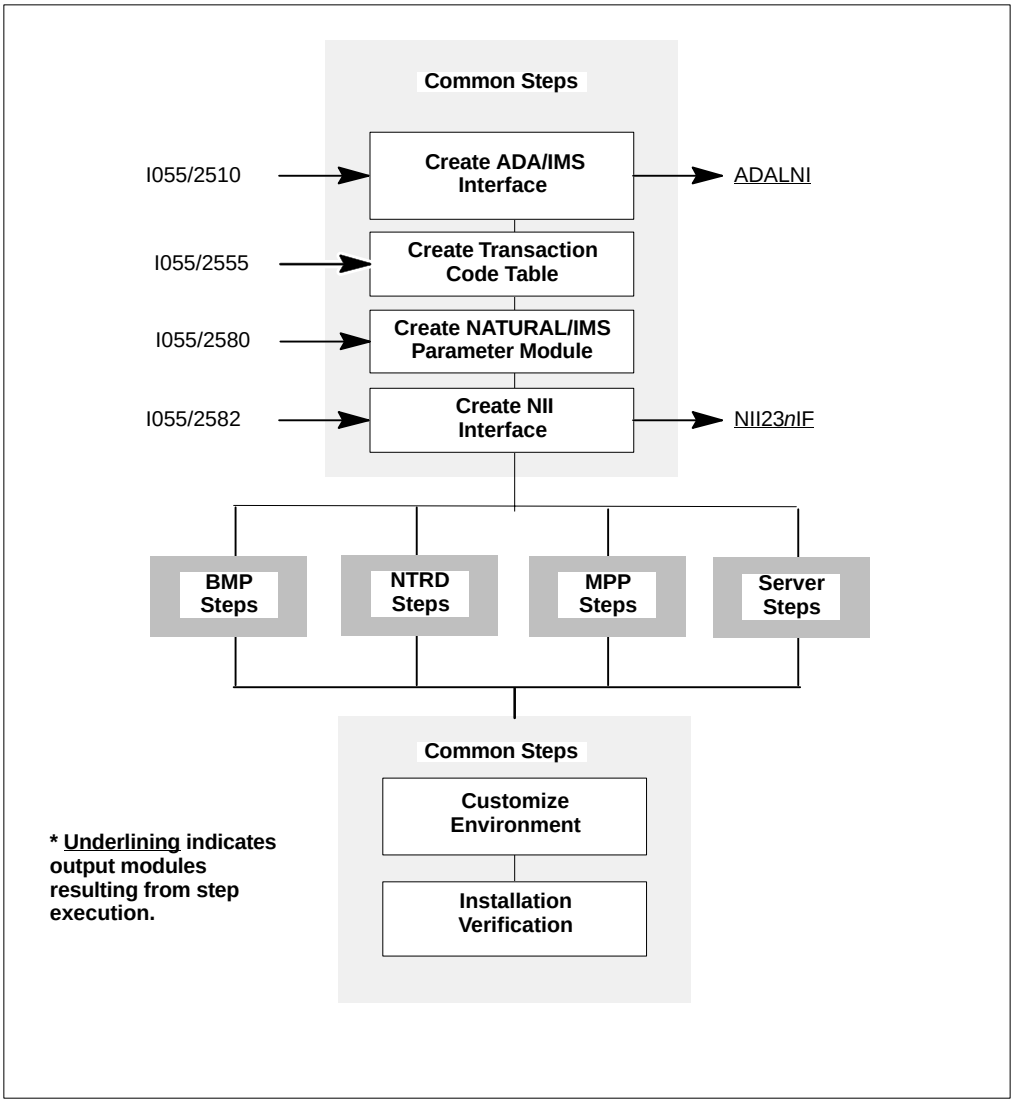
The installation procedure consists of the following steps:

1. Common installation steps,
2. Installation steps specific to a BMP environment,
3. Creating the message-oriented environment (optional),
4. Installation steps specific to an MPP environment,
5. Customizing the IMS/TM environment,
6. Creating the server environment.

Perform the steps in the sequence indicated above.

Common Installation Steps

The following steps are required to install all environments:



Step 1: Assemble and Link the ADABAS/IMS Interface (Job I055, Step 2510)

1. Modify the member ADALNI from the ADABAS/IMS source distribution library to meet your requirements. For further information, see the *ADABAS Implementation and Maintenance Manual*.

2. Assemble and link the ADABAS/IMS interface.

Step 2: Create and Assemble the NATURAL/IMS Transaction Code Table (Job I055, Steps 2555 and 2556)

1. Create the NATURAL/IMS transaction code table by including a NIMTRNTG macro for each transaction code used for NATURAL transactions.

For further information on the parameters in the NIMTRNTG macro, see page 56.

Note: If you want to use NATURAL in non-message-driven BMP or a batch environment, add a NIMTRNTG macro for the PSB used with an arbitrary transaction code.

2. Assemble and link the transaction code table.

Step 3: Create and Assemble the NATURAL/IMS Parameter Module (Job I055, Steps 2580 and 2581)

1. Create the NATURAL/IMS paramter module by including a NIMPARM macro for each environment needed.

For information on the parameters for the NIMPARM macro, see page 46.

2. Assemble and link the NATURAL/IMS parameters module.

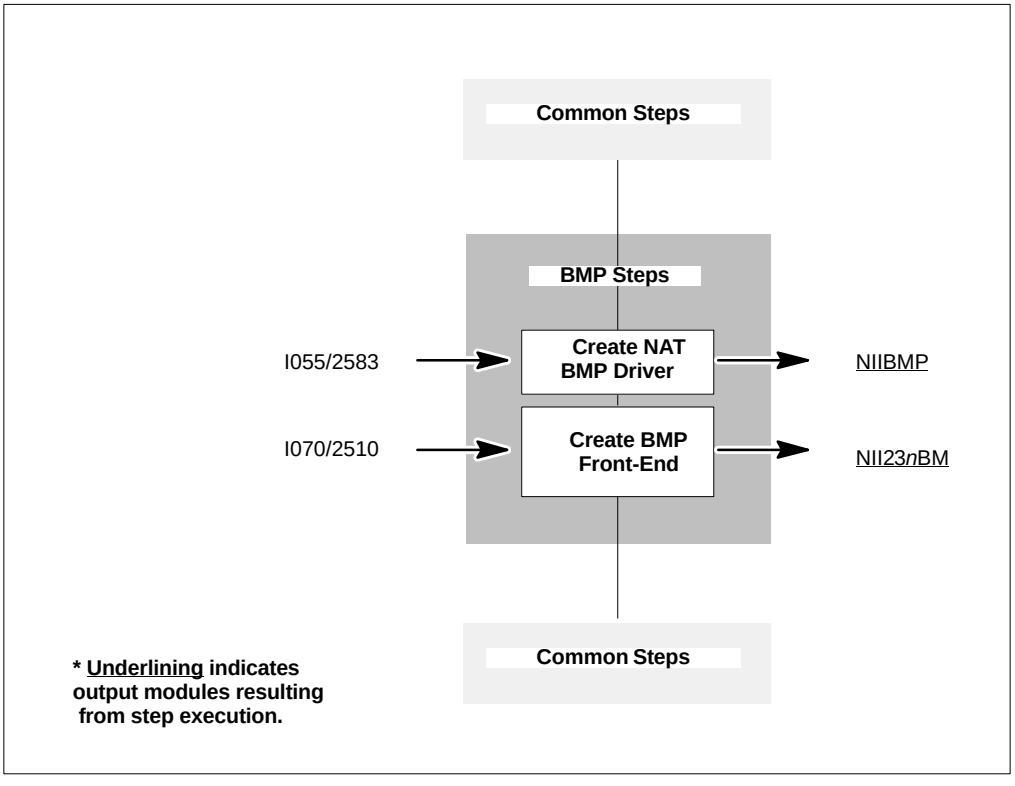
Step 4: Link the NATURAL/IMS Interface Module (Job I055, Step 2582)

1. Link the NATURAL/IMS interface module.

This module is applicable to all environments.

Installing the Batch Message Program (BMP)

The following steps are required to install the NATURAL BMP interface:



Step 1: Create the NATURAL/IMS BMP Interface (Job I055, Steps 2583 and 2584)

1. Create the source NIIBMP which contains a call to macro NIMDRIV with the parameter TYPE set to “BMP”.

For further information on the macro NIMDRIV, see page 45.

2. Assemble and link the NATURAL/BMP interface.

Note: For CMPRMTB you receive the warning IEW0461. You can ignore this.

Note: If LE370 is set to YES, you receive the warning IEW0461 for modules starting with CEE. You can ignore this.

Step 2: Link the NATURAL/IMS BMP Front-End (Job I070, Step 2510)

The front-end consists of the BMP interface created in the previous step and your batch NATURAL parameter module NATPARM.

1. Specify the name of the NATURAL batch parameter module with the INCLUDE instruction in the parameter module (Job I060, Steps 0610, 0015).

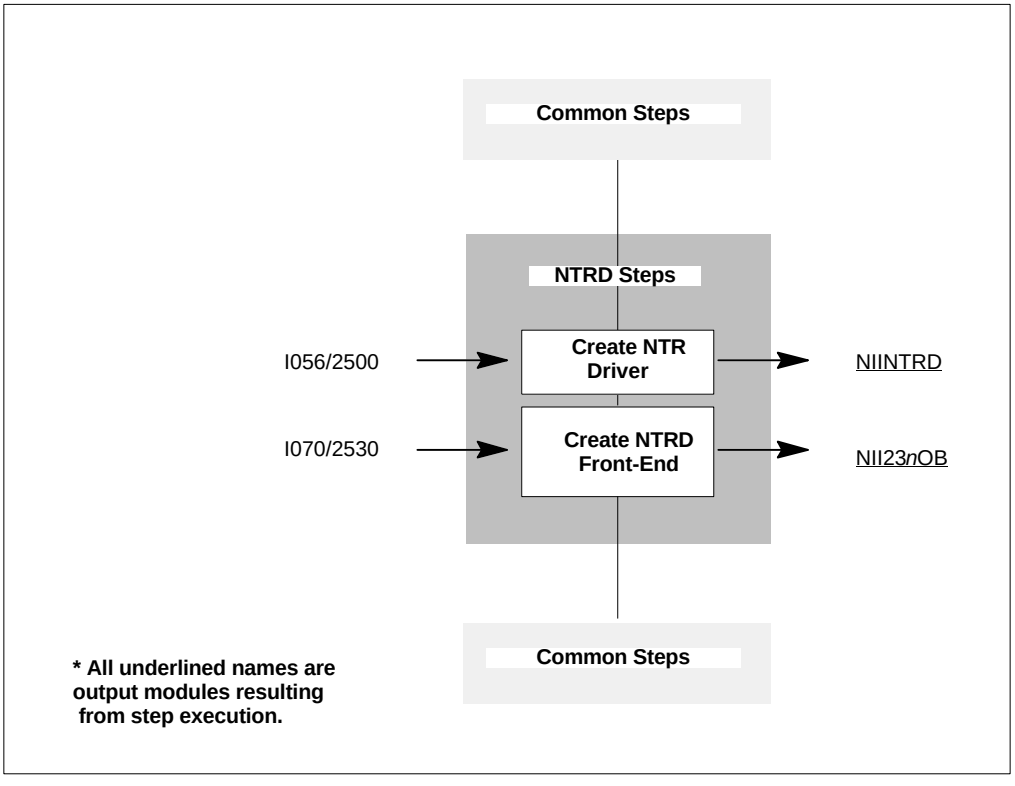
2. Specify the name of the front-end module used for this link.

Note: This name must also be specified in your BMP region job.

3. Link the front-end for the NATURAL/IMS BMP environment.

Installing the Message-Oriented NATURAL/IMS Interface

The following steps are required to install the message-oriented NATURAL/IMS interface:



Step 1: Create the NATURAL/IMS NTR Interface (Job I056, Steps 2500, 2501)

1. Create the source NIINTRD which contains a call to macro NIMDRIV with the parameter TYPE set to “NTRD”.

For further information on the macro NIMDRIV, see page 45.

2. Assemble and link the NATURAL/IMS NTRD interface.

Note: For CMPRMTB you receive the warning IEW0461. You can ignore this.

Note: If LE370 is set to YES, you receive the warning IEW0461 for modules starting with CEE. You can ignore this.

Step 2: Link the NATURAL/IMS NTR Front-End (Job I070, Step 2530)

The front-end consists of the NTR interface created in the previous step and of your batch NATURAL parameter module NATPARM.

1. Specify the name of the NATURAL batch parameter module with the INCLUDE instruction in the parameter module (Job I060, Steps 0610, 0015).

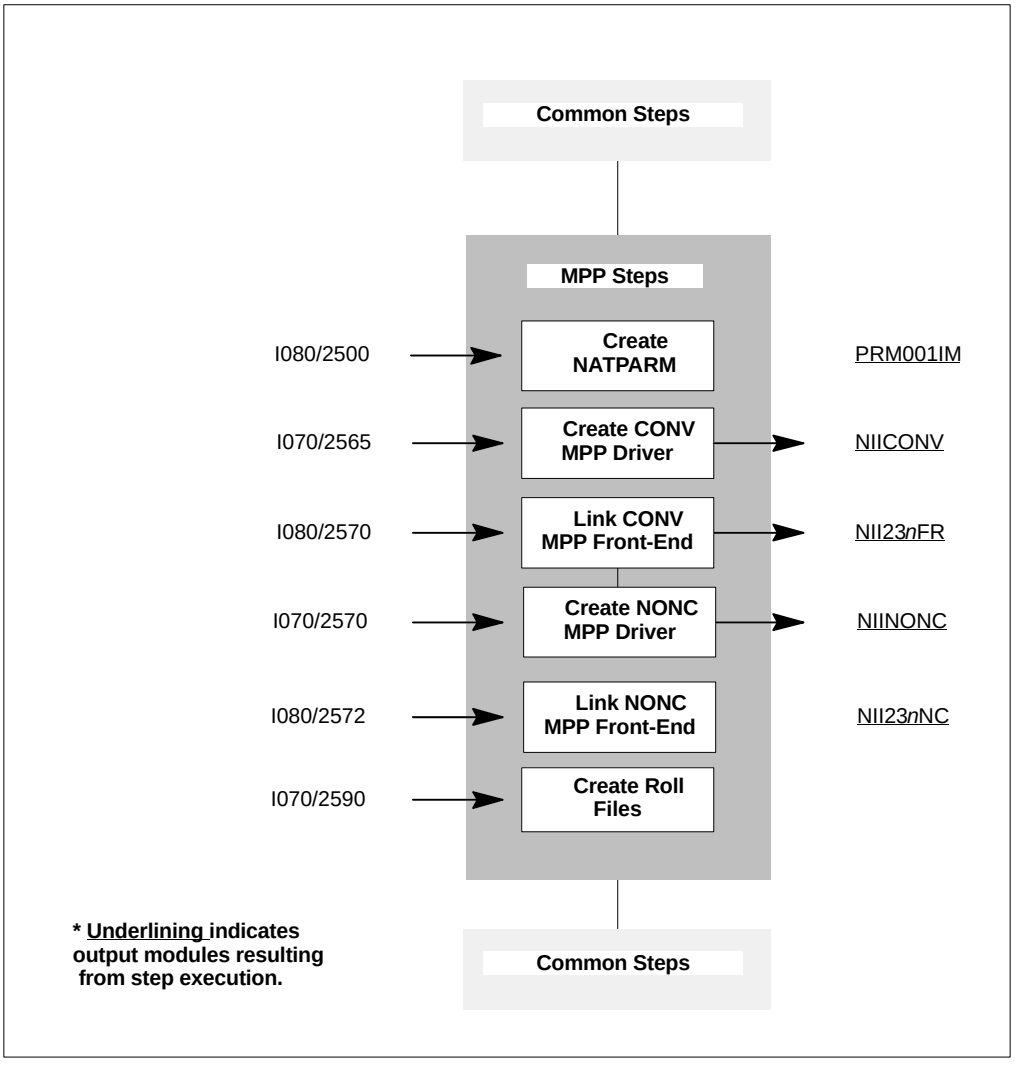
2. Specify the name of the front-end module used for this link.

Note: This name must also be specified in the APPLCNT macro.

3. Link the front-end for the NATURAL/IMS NTR interface.

Installing the Message Processing Program (MPP)

The following steps are required to install the NATURAL MPP interface:



Step 1: Create the Online NATURAL Parameter Module (Job I080, Steps 2500, 2510)

1. Set the values of the following parameters in the parameter module:

FNAT=(*dbid,fnat***)**

FUSER=(*dbid,fuser***)**

where *dbid*, *fnat*, and *fuser* are the values you specified when loading the system files in your base NATURAL installation. For further information, see the *NATURAL Installation and Operations Manual for Mainframes*.

2. To use a global buffer pool, specify the macro NTBPI in the parameter module and set the parameter SUBSID in the profile parameter module.

Note: It is strongly recommended that you use a global buffer pool.

3. Modify any other parameters in the parameter module whose default values do not meet your requirements. For further information on the parameters contained in the parameter module, see the *NATURAL Installation and Operations Manual for Mainframes*.

4. Assemble and link the NATURAL parameter module for the dialog-oriented environments.

Step 2: Create the NATURAL/IMS Conversational MPP Interface (Job I070, Steps 2565,2566)

1. Create the source NIICONV which contains a call to macro NIMDRIV with the parameter TYPE set to “CONV”.

For further information on the macro NIMDRIV, see page 45.

2. Assemble and link the NATURAL/IMS Conversational MPP interface.

Step 3: Link the NATURAL/IMS Conversational MPP Front-End (Job I080, Step 2570)

The front-end consists of the NATURAL/IMS Conversational MPP interface created in the previous step and the online NATURAL parameter module NATPARM created on page 116.

1. Specify the name of the online NATURAL parameter module in the INCLUDE instruction for the parameter module.

2. Specify the name of the front-end module used for this link.

Note: This name must also be specified in your MPP region job.

3. Link the front-end for the NATURAL/IMS Conversational MPP environment.

Step 4: Create the NATURAL/IMS Non–Conversational MPP Interface (Job I070, Steps 2570, 2571)

1. Create the source NIINONC which contains a call to macro NIMDRIV with the parameter TYPE set to “NONC”.

For further information on the macro NIMDRIV, see page 45.

2. Assemble and link the NATURAL/IMS Non-Conversational MPP interface.

Step 5: Link the NATURAL/IMS Non-Conversational MPP Front-End (Job I080, Step 2572)

The front-end consists of the NATURAL/IMS Non-Conversational MPP interface created in the previous step and the online NATURAL parameter module NATPARM created on page 116.

1. Specify the name of the online NATURAL parameter module in the INCLUDE instruction for the parameter module.

2. Specify the name of the front-end module used for this link.

Note: This name must also be specified in your MPP region job.

3. Link the front-end for the NATURAL/IMS Non-Conversational MPP environment.

Step 6: Allocate and Format the NATURAL/IMS Roll Files (Job I070, Step 2590)

Note: This step is only required if you do not use the roll server.

If you do not want to use the NATURAL roll file server, you have to allocate and format the roll files to be used by NATURAL/IMS.

You can allocate up to 5 sequential datasets with a fixed-record format for use as roll files.

1. Allocate the roll files.

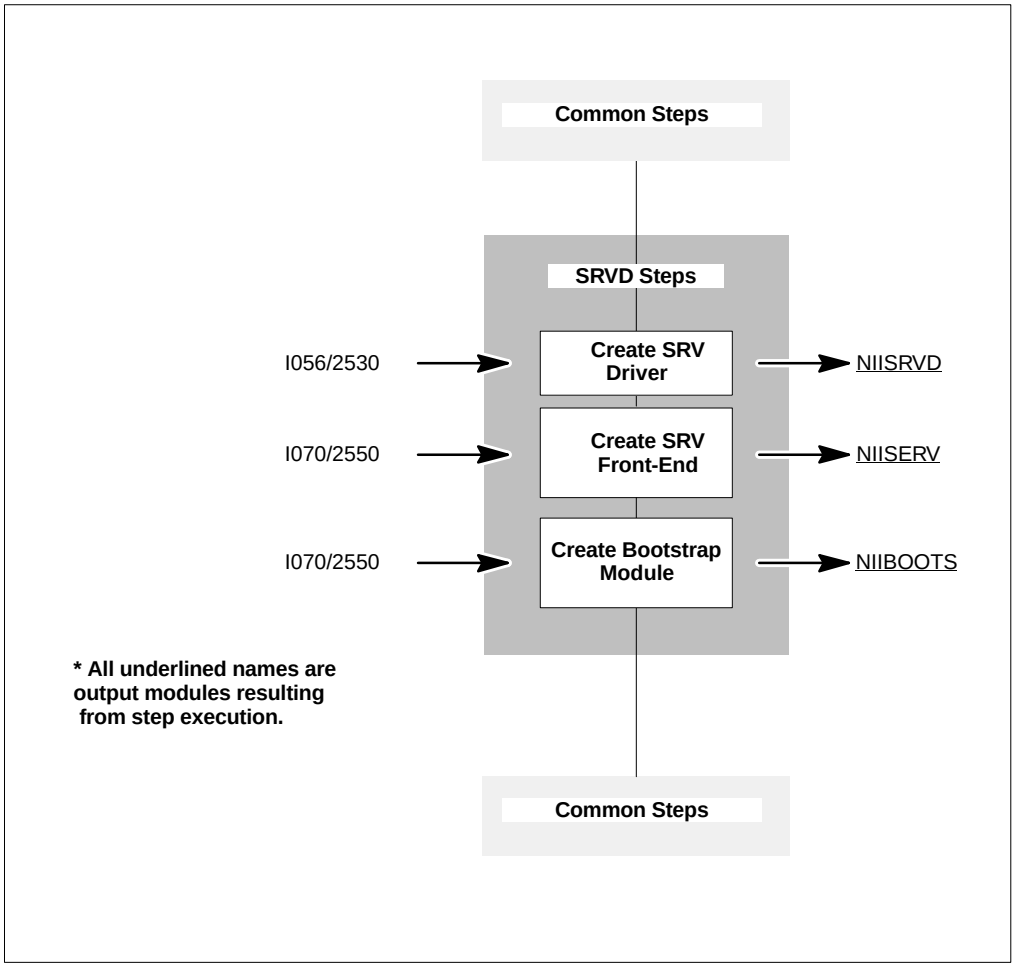
2. Format the roll files using the module NATRSRFI.

The roll file initialization program produces a WTO message indicating the number of concurrent users which can be serviced by the roll file.

For information on the roll file facility, see the *NATURAL Installation and Operations Manual for Mainframes*.

Installing the NATURAL/IMS Server Interface (SRVD)

The following steps are required to install the NATURAL/IMS server interface:



Step 1: Create the NATURAL/IMS SRV Interface (Job I056, Steps 2530, 2531)

1. Create the source NIISRVD which contains a call to macro NIMDRIV with the parameter TYPE set to “SRVD”.

For further information on the macro NIMDRIV, see page 45.

2. Assemble and link the NATURAL/IMS SRV interface.

Note: For CMPRMTB you receive the warning IEW0461. You can ignore this.

Note: If LE370 is set to YES, you receive the warning IEW0461 for modules starting with CEE. You can ignore this.

Step 2: Link the NATURAL/IMS SRV Front-End (Job I070, Step 2550)

The front-end consists of the SRV interface created in the previous step and of your batch NATURAL parameter module NATPARM.

1. Specify the name of the NATURAL batch parameter module with the INCLUDE instruction in the parameter module (Job I060, Steps 0610, 0015).

2. Specify the name of the front-end module used for this link.

Note: This name must also be specified in the NIMBOOT macro.

3. Link the front-end for the NATURAL/IMS SRV interface.

Step 3: Create the Bootstrap Module NIIBOOTS (Job I056, Steps 2540, 2541)

1. Create the source NIIBOOTS which contains a call to macro NIMBOOT with the parameter SERVER set to “YES”. For the DRIVERN parameter, specify the name of the SRV front-end module created in step 2.

2. Assemble and link the NATURAL/IMS bootstrap module.

Customizing your IMS Environment

The following steps require system modifications to your IMS environment.

Step 1: Create the APPLCTN Table Definitions for MPP, BMP and NTR

MPP Define Sample:

```
APPLCTN PSB=NII23nFR,PGMTYPE=TP
    TRANSACT  CODE=NAT23n,MODE=SNGL,SPA=512,
        MSGTYPE=(SNLGSEG,RESPONSE,9)
```

Note: The size of the SPA must be set to at least 173 bytes.

BMP Define Sample (Message-Driven or NAF-specific BMP):

```
APPLCTN PSB=NII23nBM,PGMTYPE=BATCH
    TRANSACT  CODE=NATBMP,MODE=SNGL,
        MSGTYPE=(SNLGSEG,RESPONSE,9)
```

Note: This APPLCTN definition is required if you use the CMGETMSG feature.

BMP Define Sample (without Message Queue Processing):

```
APPLCTN PSB=NII23nBM,PGMTYPE=BATCH
```

NTR Define Sample:

```
APPLCTN PSP=NII23nOB,PGMTYPE=TP
    TRANSACT  CODE=NATOBMP,MODE=SNGL,
        MSGTYPE=(MULTSEG,NONRESPONSE,10)
```

Step 2: Create the PSB/ACB for both the MPP and BMP

Example for MPP:

PSB for conversational NATURAL:

```
PCB    TYPE=TP,MODIFY=YES
PCB    TYPE=TP,MODIFY=YES
PCB    TYPE=TP,MODIFY=YES
PCB     TYPE=DB,DBNAME=dliddm,PROCOPT=A,KEYLEN=16    sample for NDL
SENSEG NAME=EMPLOY,PROCOPT=A                         sample for NDL
SENSEG NAME=VEHICL,PROCOPT=A,PARENT=EMPLOY           sample for NDL
```

MPP PSB definition:

```
PSBGEN   PSBNAME=NII23nFR,LANG=ASSEM,MAXQ=3,IOASIZE=132
```

At least one modifiable TP-PCB must be defined for default use of hardcopy, sending messages and transaction switching. To avoid a NATURAL initialization error, the value of the WRKPCBS parameter in the current environment table must be lesser than or equal to the number of PCBs minus 1.

Example for BMP:

```
PCB    TYPE=TP,MODIFY=YES
PCB    TYPE=TP,MODIFY=YES
PCB    TYPE=DB,DBNAME=dliddm,PROCOPT=A,KEYLEN=16     sample for NDL
SENSEG NAME=EMPLOY,PROCOPT=A                         sample for NDL
SENSEG NAME=VEHICL,PROCOPT=A,PARENT=EMPLOY           sample for NDL

PSBGEN PSBNAME=NII23nBM,LANG=ASSEM,MAXQ=3,IOASIZE=132
```

At least one modifiable TP-PCB must be defined for default use of hardcopy, sending messages and transaction switching. To avoid a NATURAL initialization error, the value of the WRKPCBS parameter in the current environment table must be lesser than or equal to the number of PCBs minus 1.

After creating the required APPLCTNs for the BMP and MPP environments, you must generate the PSB, DBD and ACB.

After the ACB generation, the following commands activate the new definitions:

```
MODIFY PREP ACBLIB
MODIFY COMMIT
```

Step 3: Create the BMP and MPP Regions

Use the sample members as guidelines when creating the specific regions for your installation.

```
BMPJOB
MPPJOB
```

Step 4: Create the PRELOAD List

Update the PRELOAD list using a PRELOAD member DFSMPL*xx* with the following module names:

```
the NATURAL nucleus name,
the name of the NII interface,
the front-end name,
the ADABAS link module name
```

Example for MPP:

```
NATUR23,NII23nIF,NII23nFR,ADALNI
```

Example for BMP:

```
NATUR23,NII23nIF,NII23nBM,ADALNK
```

If ALIAS names are used for any members in the PRELOAD list, these names should be added to the PRELOAD list as well. Failure to do so leads to performance degradation.

Special Considerations

- Set the REGION parameter to at least 2 MB.
- Include LOAD libraries used to create the NATURAL/IMS environment.
- Include the DD statement for the roll file created in job I070M, step 2590:

  ```
  //ROLLFn   DD  DSN=....DISP=SHR
  ```

  where *n* is a value from 1 – 5.
- Include the DD statement for NATRJE:

  ```
  //NIIRJEDD   DD  SYSOUT=(X,INTRDR)
  ```

Installing the Multi-Session Feature (Optional)

The Multi-Session Feature is an optional feature. For further information, see the section **The Multi-Session Feature** (page 98).

Step 1: Create the Multi-Session Database

The following steps have no corresponding example jobs in NAT*nnn*.JOBS.

1. Create the DBD using the member NIIMSDBD in NII23*n*.SRCE.
2. Create the PSB for the initial load.
3. Add the DBD to all PSBs intended for use with the multi-session feature.
4. Define and load the database using the JCL INITDM in NAT23*n*.JOBS.

Step 2: Adapt the NII Parameter Module (Environment Table)

Parameter	Description
MSACTV=YES	Activates the session manager
MSDBD=*dbdname*	Where *dbdname* is the name used in MSDBD
MSCRKEY	The key to create a new session
MSRSKEY	The key to switch to a resumed session
MSMAX=*nn*	Where *nn* is the number of active sessions (max. 8)

For further information, see the sections **Installing the Batch Message Program (BMP)** (page 111), and **Installation Steps for the Message Processing Program (MPP)**, (page 115).

Step 3: Adapt the Transaction Code Table

Set the following parameter:

Parameter	Description
MSPCB	Number of the multi-session PCB

Installation Verification

1. From an IMS session, start the BMP with the following command:

```
/STA REG BMPJOB
```

2. Check output.

The output results from the NATURAL system command TECH. Verify the output with your environment.

3. Issue the following IMS commands from the IMS session:

```
/STA REG MPPJOB
/STA TRAN NAT23n
/STA PROG NII23nFR
```

4. Type in transaction name NAT23*n*.

5. Proceed with the steps described in the section **Installation Verification for TP Monitor Interface** in the *NATURAL Installation and Operations Manual for Mainframes*.

Error Codes

This section lists the error codes and messages that may be issued by the NATURAL/IMS Interface (NII).

Note: Error texts are not issued for errors of the types "abend" and "abend with dump".

3500

Type	Abend with dump
Issuing Module	NIICONV/NIINONC/NIINTRD
Reason	You are not allowed to execute a driver of type CONV, NONC or NTRD in a BMP region.
Action	Check your installation.

3501

Type	Abend with dump
Issuing Module	NIIBMP
Reason	You are not allowed to execute a driver of type BMP in an MPP region.
Action	Check your installation.

3502

Type	Abend with dump
Issuing Module	NIICONV/NIINONC
Reason	The GU against the message queue was not successful.
Action	Check the status code contained in register 15.

3503

Type	Abend with dump
Issuing Module	All drivers
Reason	Unable to load the NATURAL/IMS interface module.
Action	Check the NIINAME parameter of the driver and/or the steplib concatenation of the region.

3504

Type	Error
Issuing Module	NIIINITS/NIIULDSN
Reason	Unable to load the NATURAL nucleus.
Action	Check your NATURAL profile parameter NUCNAME and/or the steplib concatenation of your region.

3505

Type	Abend with dump
Issuing Module	All drivers.
Reason	The entry within the transaction code table could not be located.
Action	Check your transaction code table.

3506

Type	Abend with dump
Issuing Module	All drivers.
Reason	The entry within the NII parameter module could not be located.
Action	Check the transaction code table and/or environment table.

3507

Type	Error
Issuing Module	NIICONV/NIINONC
Reason	The GN against the message queue was not successful.
Action	Check the status code contained in the message.

3508

Type	Error
Issuing Module	NIICONV/NIINONC
Reason	The received input message is greater than the input message buffer.
Action	Increase the size of the input message buffer in your environment table.

3509

Type	Error
Issuing Module	NIIINITS
Reason	The session start exit NIIXSSTA requested that the NATURAL session be terminated abnormally.
Action	Check the return code contained in the message.

3510

Type	Abend with dump
Issuing Module	NIICONV/NIINONC
Reason	The ISRT of the termination message was not successful.
Action	Check the status code contained in register 15.

3511

Type	Abend with dump
Issuing Module	NIICONV
Reason	The ISRT of the SPA was not successful.
Action	Check the status code contained in register 15.

3513

Type	Abend with dump
Issuing Module	All drivers
Reason	The storage for the working area, the buffers or the NATURAL thread could not be obtained.
Action	Check the reason code contained in register 15 and/or increase the size of your region.

3514

Type	Error
Issuing Module	NIIBMP
Reason	Mandatory DD statement missing.
Action	Check your BMP JCL for the DD name contained in the message.

3515

Type	Error with dump
Issuing Module	NIIBMP
Reason	Open failed.
Action	Check your BMP JCL for the DD name contained in the message.

3517

Type	Error
Issuing Module	NIIOLDSN
Reason	The start exit NIIXSTAR requested that the NATURAL session be terminated abnormally.
Action	Check the return code contained in the message.

3518

Type	Abend with dump
Issuing Module	NIISRTQ
Reason	The CHNG call for the printer destination specified with the NATURAL profile parameter SENDER failed.
Action	Check the status code contained in register 15. Probably invalid printer destination.

3519

Type	Abend with dump
Issuing Module	NIIISRTQ
Reason	The ISRT call to the assigned printer destination failed.
Action	Check the status code contained in register 15.

3520

Type	Abend with dump
Issuing Module	NIIISRTQ
Reason	The PURG call for the assigned printer destination failed.
Action	Check the status code contained in register 15.

3521

Type	Abend with dump
Issuing Module	NIIBMP
Reason	Either NATURAL terminated with a bad return code or an NII error occurred and you requested an abend.
Action	If the abend is caused by a NATURAL error, CMPRINT contains the error message. If the abend is caused by a NATURAL/IMS error, a WTO with the error message has been issued before.

3522

Type	Error
Issuing Module	NIICONV/NIINONC
Reason	The SPA is greater than the size specified for the SPA buffer.
Action	Check the environment table.

3523

Type	Error
Issuing Module	NIIBMP
Reason	The storage for the back-end program area could not be allocated.
Action	Check the TERMINATE statement and/or increase the size of your BMP region.

3524

Type	Error with dump
Issuing Module	NIIBMP
Reason	The back-end program could not be loaded.
Action	Check the name of the back-end program contained in the message and/or the steplib concatenation of your region.

3525

Type	Abend with dump
Issuing Module	NIIDRIV
Reason	The save area stack is exhausted.
Action	Contact SOFTWARE AG support.

3527

Type	Error
Issuing Module	NIICONV, NIINONC
Reason	You specified RELO=FORCE, but there is insufficient memory for a new thread.
Action	Increase the size of your region.

3530

Type	Abend
Issuing Module	NIIBOOTS
Reason	Unable to obtain our name/token pair.
Action	Check the return code contained in register 15 and/or the steplib concatenation of your region.

3531

Type	Error
Issuing Module	NIIBOOTS
Reason	Unable to load the driver.
Action	Check the return code contained in register 15 and/or the steplib concatenation of your region.

3600

Type	Error
Issuing Module	NIIVTRTA
Reason	Transaction type does not match driver type. E.g. you specified transaction code conversational in the transaction code table but the driver type is non-conversational and vice versa.
Action	Check the transaction code table.

3601

Type	Error
Issuing Module	NIIVTRTA
Reason	The SPA is too small. The SPA must be at least 157 bytes plus the offset of the NATURAL reserved area defined in the transaction table.
Action	Check the transaction code table and/or increase the size of the SPA.

3602

Type	Error
Issuing Module	NIIVTRTA
Reason	Alternate TP PCB missing. At least one modifiable alternate TP PCB is required.
Action	Check your PSB.

3604

Type	Error
Issuing Module	NIIVTRTA
Reason	Not enough alternate TP PCBs. The PSB does not contain the number of alternate modifiable TP PCBs specified in the transaction code table as additional work PCBs.
Action	Check your transaction code table and/or your PSB.

3605

Type	Error
Issuing Module	NIIVTRTA
Reason	Message PCB is not a modifiable alternate TP PCB. You specified a separate message PCB in the transaction code table, but your PSB does not contain enough modifiable alternate TP PCBs.
Action	Check your transaction code table and/or your PSB.

3607

Type	Error with dump
Issuing Module	NIIVTRTA
Reason	The PCB number specified for the logical PCB name is greater than the maximum number of PCBs.
Action	Check your transaction code table and/or your PSB for the PCB name contained in the message.

3611

Type	Error
Issuing Module	NIIINITS
Reason	The user exit NIIXRFNU returned a roll file number not in the range of 1 to the total available number of roll files as defined in the environment table, that is, larger than the total number of allowed roll files.
Action	Correct the user exit.

3612

Type	Error with dump
Issuing Module	NIIINITS
Reason	The roll server ALLOC request failed. The message contains the returncode/reason code.
Action	Contact SOFTWARE AG support.

3613

Type	Error
Issuing Module	NIIINITS
Reason	The roll server ALLOC request failed because the roll file is full.
Action	Increase the size of the roll file or increase the number of roll files.

3614

Type	Error
Issuing Module	NIIINITS,NIIOLDSN,NIIWTERM,NIIENDSN
Reason	The roll server is not active.
Action	Check the roll server status and restart the roll server.

3615

Type	Error
Issuing Module	NIIINITS/NIIOLDSN
Reason	The loaded NATURAL nucleus is invalid.
Action	Specify a correct NATURAL nucleus name and/or check the steplib concatenation of your region.

3616

Type	Error
Issuing Module	NIIINITS/NIIOLDSN
Reason	The loaded NATURAL nucleus is not of Version 2.3.
Action	Specify a correct NATURAL nucleus name and/or check the steplib concatenation of your region.

3617

Type	Error
Issuing Module	NIIINITS/NIIOLDSN
Reason	The NATURAL parameter module is not linked to the NATURAL/IMS front-end.
Action	Check the result of your front-end’s link-edit.

3620

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	Error during address relocation. Register 15 contains the NATRELD return code.
Action	Contact SOFTWARE AG support.

3621

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	An IPL has occurred since the start of the NATURAL session.
Action	Restart your session.

3622

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	Error during decompression. The message contains the NATCODEC return code.
Action	Contact SOFTWARE AG support.

3624

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	A wrong user thread has been rolled in.
Action	Contact SOFTWARE AG support.

3626

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	The roll server READ request failed because the roll file and/or the roll server have been reinitialised.
Action	Restart the NATURAL session.

3627

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	The roll server READ request failed. The message contains the return code/reason code.
Action	Contact SOFTWARE AG support.

3628

Type	Error with dump
Issuing Module	NIIOLDSN
Reason	You cancelled the session by entering the cancel string as specified in the NATURAL profile parameter CANCEL.
Action	Usual dump procedure

3630

Type	Error
Issuing Module	NIIMSESS
Reason	The MSPCB specified in the transaction code table does not address the multiple–session database.
Action	Check your transaction code table and/or your PSB.

3631

Type	Error with dump
Issuing Module	NIIMSESS
Reason	Access to the multiple- session database failed.
Action	Check the status code contained in the message.

3632

Type	Error with dump
Issuing Module	NIIMSESS
Reason	The entry for the transaction code to be resumed could not be found in the transaction code table.
Action	Contact SOFTWARE AG support.

3633

Type	Error with dump
Issuing Module	NIIMSESS
Reason	The ISRT call for the SPA or the message failed.
Action	Check the status code contained in the message.

3634

Type	Error with dump
Issuing Module	NIIMSESS
Reason	The CHNG call for the transaction code to be resumed failed.
Action	Check the status code contained in the message and/or contact SOFTWARE AG support.

3635

Type	Error with dump
Issuing Module	NIIMSSPF
Reason	The multiple-session prefix could not be built due to an illegal character in the terminal name. Register 2 of the error registers in the dump points to the illegal character.
Action	Check the terminal name and/or use the NIIXMSSP exit.

3636

Type	Error with dump
Issuing Module	NIIMSSPF
Reason	An invalid return code was received from the NIIXMSSP exit. Register 2 of the error registers in the dump contains the return code.
Action	Check your exit.

3640

Type	Error with dump
Issuing Module	NATAM0G
Reason	Unexpected status code for the CHNG call.
Action	Check the status code contained in the message and/or contact SOFTWARE AG support.

3641

Type	Error with dump
Issuing Module	NATAM0G
Reason	Unexpected status code on the ISRT call.
Action	Check the status code contained in the message and/or contact SOFTWARE AG support.

3642

Type	Error with dump
Issuing Module	NATAM0G
Reason	Unexpected status code on the PURG call.
Action	Check the status code contained in the message and/or contact SOFTWARE AG support.

3650

Type	Error
Issuing Module	NIICDBPC
Reason	Database not available.
Action	Start the database.

3651

Type	Error
Issuing Module	NIICDBPC
Reason	The PROCOPT option in the PCB is set to update but the database is not available for update.
Action	Check the database.

3652

Type	Abend
Issuing Module	NIIINQYE
Reason	The INQY ENVIRON call failed. Register 15 contains the return code in bytes 0 and 1 and the reason code in bytes 2 and 3.
Action	Contact SOFTWARE AG support.

3660

Type	Error
Issuing Module	NIIWTERM
Reason	The compression routine returned a bad return–code.
Action	Check the return code in Register 15 and contact SOFTWARE AG support.

3661

Type	Error with dump
Issuing Module	NIIWTERM
Reason	The routine NATRELSE failed.
Action	Contact SOFTWARE AG support.

3662

Type	Error with dump
Issuing Module	NIIWTERM
Reason	The ISRT call to the alternate PCB for the SPA failed.
Action	Check the status code in Register 15.

3663

Type	Error with dump
Issuing Module	NIIWTERM
Reason	The ISRT call to the alternate PCB for the message failed.
Action	Check the status code in Register 15.

3664

Type	Abend with dump
Issuing Module	NIIWTERM
Reason	The ISRT call for the SPA failed.
Action	Check the status code in Register 15.

3665

Type	Abend with dump
Issuing Module	NIIWTERM
Reason	The ISRT call for the message failed.
Action	Check the status code in Register 15.

3666

Type	Error
Issuing Module	NIIWTERM
Reason	The current slot size exceeds the maximum slot size. The message contains the slot size required in blocks for the roll file.
Action	Increase the slot size of your roll file.

3668

Type	Error with dump
Issuing Module	NIIWTERM
Reason	Roll server WRITE request failed. The message contains the return code/reason code.
Action	Contact SOFTWARE AG support.

3669

Type	Error
Issuing Module	NIIWTERM
Reason	You tried to switch to a non-NATURAL session using a NATURAL transaction code.
Action	Correct the application.

3670

Type	Error
Issuing Module	NIIWTERM
Reason	You can not switch directly from a non-conversational transaction code to a conversational one.
Action	Correct the application.

3671

Type	Error with dump
Issuing Module	NIIWTERM
Reason	The CHNG call for the transaction switch returned a bad status code. The message contains the first five bytes of the transaction code and the status code.
Action	Check the status code and/or correct the application.

3680

Type	Error
Issuing Module	NIISIPSS
Reason	The subsystem entry for SPATID could not be located. The message contains a subsystem ID as specified in the environment table.
Action	Check your subsystem definitions and/or your environment table.

3682

Type	Error
Issuing Module	NIISIPSS
Reason	The session information pool (SIP) is full.
Action	Restart the Authorised Services Manager with an increased number of SIP slots. For further information, see the *NATURAL Installation and Operations Manual for Mainframes*.

3683

Type	Error
Issuing Module	NIISIPSS
Reason	The Authorised Services Manager is not active. The Authorised Services Manager must be active if the terminal-oriented non-conversational environment is used.
Action	Start the Authorised Services Manager. For further information, see the *NATURAL Installation and Operations Manual for Mainframes*.

3684

Type	Error
Issuing Module	NIISIPSS
Reason	The SIP function of the Authorised Services Manager is disabled. The SIP function must be active if the terminal-oriented non-conversational environment is used.
Action	Start the Authorised Services Manager with the SIP function enabled as described in the *NATURAL Installation and Operations Manual for Mainframes*.

3685

Type	Error with dump
Issuing Module	NIISIPSS
Reason	The Authorised Services Manager SIP function issued an unexpected return code. The message contains the return and the reason code.
Action	For further information on the return/reason codes see the chapter **Authorised Services Manager** in the *NATURAL Installation and Operations Manual for Mainframes*. If necessary, contact SOFTWARE AG support.

3686

Type	Error
Issuing Module	NIISIPSS
Reason	The size of the SPA buffer is greater than the SIP slot size.
Action	Increase the SIP slot size as described in the chapter **Authorised Services Manage** in the *NATURAL Installation and Operations Manual for Mainframes*.

3690

Type	Error
Issuing Module	NIIINITS, NIIOLDSN, NIIWTERM
Reason	The subsystem entry could not be located. The message contains the subsystem ID as specified with the NATURAL profile parameter SUBSID.
Action	Check your subsystem definitions and/or your environment table.

3692

Type	Error with dump
Issuing Module	NIIENDSN
Reason	The roll server FREE request failed. The message contains the return code/reason code.
Action	Contact SOFTWARE AG support.

3800

Type	Error with dump
Issuing Module	NIIBDAMR
Reason	DCB not allocated for one of the roll files. The message contains the DD name of the roll file.
Action	Contact SOFTWARE AG support.

3801

Type	Error
Issuing Module	NIIBDAMR
Reason	The OPEN of the roll file failed.
Action	Check your region JCL for the DD name contained in the message.

3802

Type	Error with dump
Issuing Module	NIIBDAMR
Reason	Invalid function code for NIIBDAMR.
Action	Contact SOFTWARE AG support.

3803

Type	Error
Issuing Module	NIIBDAMR
Reason	No free slot in roll file. The message contains the DD name.
Action	Increase the size of your roll file.

3804

Type	Error
Issuing Module	NIIBDAMR
Reason	The roll file has been reformatted since the start of the session.
Action	Restart the session.

3805

Type	Error
Issuing Module	NIIBDAMR
Reason	The intermediate buffer for the roll slot could not be obtained.
Action	Increase the region size.

3810

Type	Error with dump
Issuing Module	NIISEOS
Reason	The CHNG failed.
Action	Check the status code contained in the message.

3811

Type	Error
Issuing Module	NIISEOS
Reason	You tried to switch from a non-conversational NATURAL session to a conversational one.
Action	Check your application.

3812

Type	Error with dump
Issuing Module	NIISEOS
Reason	The ISRT call for the message into the alternate PCB failed.
Action	Check the status code contained in the message.

3813

Type	Error with dump
Issuing Module	NIISEOS
Reason	The ISRT call for the SPA into the alternate PCB failed.
Action	Check the status code contained in the message.

3814

Type	Error
Issuing Module	NIISEOS
Reason	The ISRT call for the dummy message failed.
Action	Check the status code contained in the message.

3815

Type	Abend with dump
Issuing Module	NIISEOS
Reason	The ISRT call for the SPA into the IOPCB failed.
Action	Check the status code contained in Register 15.

3820

Type	Error
Issuing Module	NIISIPSM
Reason	The Authorized Services Manager is not active. The Authorized Services Manager must be active if the monitoring function is used.
Action	Start the Authorized Services Manager as indicated in the *NATURAL Installation and Operations Manual for Mainframes.*

3821

Type	Error
Issuing Module	NIISIPSM
Reason	The SIP function of the Authorized Services Manager is disabled. The SIP function must be active if the monitoring function is used.
Action	Start the Authorized Services Manager with the enabled SIP function as indicated in the *NATURAL Installation and Operations Manual for Mainframes.*

3822

Type	Error with dump
Issuing Module	NIISIPSM
Reason	An unexpected return code was received from the SIP function of the Authorized Services Manager. The message contains the return code and the reason code.
Action	For the return codes, see the *NATURAL Installation and Operations Manual for Mainframes*. If necessary, contact SOFTWARE AG support.

3823

Type	Error
Issuing Module	NIISIPSM
Reason	The SIP server pool is full.
Action	Increase the number of SIP slots in the Authorized Services Manager as indicated in the *NATURAL Installation and Operations Manual for Mainframes.*

3830

Type	Abend with dump
Issuing Module	NIIACT
Reason	The CMD command failed when issuing the /LOG command to write the accounting record. The transaction code must be authorized to use the /LOG command when ACTLOG=LOG is set for accounting.
Action	Check the status code contained in register 15.

3831

Type	Abend with dump
Issuing Module	NIIACT
Reason	The /LOG command failed when writing the accounting record.
Action	Check the status code contained in register 15.

3832

Type	Abend with dump
Issuing Module	NIIACT
Reason	The Authorized Services Manager failed when writing the accounting record to SMF.
Action	Check the status/reason code contained in register 15.

3950

Type	Error with dump
Issuing Module	NIIESTAE
Reason	A system abend code has been intercepted.
Action	Examine the dump.

3951

Type	Error with dump
Issuing Module	NIIESTAE
Reason	A recursive system abend code has been intercepted.
Action	Examine the dump.

3960

Type	Error
Issuing Module	CMGETMSG,NIGETMSG
Reason	The retrieved message exceeds the specified size of the input message buffer.
Action	Increase the size of your input message buffer.

3961

Type	Error
Issuing Module	CMGETSEG,CMGSEGO
Reason	The segment received does not fit into the input message buffer.
Action	Increase the size of the input message buffer.

3962

Type	Abend
Issuing Module	CMSVC13D, NIIU3962
Reason	The session has been abended on request.
Action	Contact the application developer.

3963

Type	Error
Issuing Module	CMSTCKSY
Reason	The synchronized STCK value could not be obtained.
Action	Contact the system administrator.

3970

Type	Error
Issuing Module	NIIOLDSN
Reason	COMPOSE has issued a terminal I/O and the NATURAL nucleus has been displaced.
Action	When using COMPOSE, put the NATURAL nucleus into the (E)LPA or use a dedicated region and put the NATURAL nucleus into the preload list.

3971

Type	Error
Issuing Module	NIIOLDSN
Reason	COMPOSE has issued a terminal I/O and the NATURAL thread has been displaced.
Action	When using COMPOSE, use a dedicated region and do not specify RELO=FORCE.

INDEX (For Chapter 2 Only)

A

Accounting, 94
ASMTDLI, DL/I language interface, 44

B

Bootstrap module, 38
Broadcasting, 97
Buffer pool, under IMS/TM, 64

C

CMCMMND load module, 79
CMDEFSW load module, 80
CMDEFSWX load module, 80
CMDIRNMX load module, 80
CMDIRNMZ load module, 81
CMDIRSWX load module, 81
CMDIRSWZ load module, 82
CMDISPCB load module, 83
CMEMOD load module, 83
CMGETMSG load module, 84
CMGETSEG load module, 84
CMGETSPA load module, 85
CMIMSID load module, 85
CMIMSINF load module, 86
CMPCBADR load module, 86
CMPRNTR load module, 87
CMPUTMSG load module, 87
CMPUTSPA load module, 88
CMQTRAN load module, 88
CMQUEUE load module, 89
CMQUEUEX load module, 89
CMSNFPRT load module, 90
CMSVC13D load module, 90
CMTRNSET load module, 91

I

Installation, of NATURAL/IMS interface, 106

M

Message text module, NIMMSGT macro,
  parameters, 60
Monitoring, 95
Multi-Session feature, 98

N

NATURAL/IMS interface
  BMP region, 34, 39
  dialog-oriented, 33, 34
  installation of, 106
  message-oriented, 34, 35
  recovery handling, 105
  service modules, 79
NIIBOOT, bootstrap module, 38
NIIBOOTS, call interface, 100
NIIINTFM, NII interface module, 43
NIIMSGT
  See also Message text module
  NII message text module, 44
NIIPARM, NII parameter module, 43
NIITRTAB, NII transaction code table, 43
NIMBOOT macro, 38
NIMDRIV macro, 42
  parameters, 45
NIMLPCB macro, parameters, 59
NIMMSGT macro, parameters, 60
NIMPARM macro, parameters, 46
NIMPIXT macro, 61
NIMTRNTG macro, parameters, 56

P

Print driver, 52

R

Roll files, under IMS/TM, 63

S

Server environment, 100
Special functions, 94